

# **XML–basierte Generierung individueller Terminpläne**

## **Diplomarbeit**

im Fach Informationsnetze, Kommunikationstechnik und  
Netzwerkmanagement

Studiengang Informationsmanagement der  
Fachhochschule Stuttgart – Hochschule der Medien

**Sebastian Spaleck**

eingereicht bei                      Prof. Dr. Wolf-Fritz Riekert  
   Prof. Dr. Christian Rathke (Zweitprüfer)

Bearbeitungszeitraum: 1. August bis 5. November 2001

Stuttgart, November 2001

## Kurzfassung

Die vorliegende Arbeit beschreibt die Entwicklung und Implementierung eines Terminauskunftssystems auf Basis von XML. Die Umsetzung erfolgt am Beispiel eines Stundenplansystems für den Fachbereich Information und Kommunikation der Hochschule der Medien, Stuttgart.

Gegenüber der gegenwärtig verbreiteten Darstellung der Daten durch HTML bringt die Beschreibung mit XML im Zusammenspiel mit XSLT den Vorteil, dass die Termine in unterschiedlicher Weise sortiert und dargestellt werden können. Somit ist eine Generierung persönlicher Terminpläne für unterschiedliche Gruppen und Personen möglich.

Die Verwendung einer XML-Darstellung des Stundenplans muss nicht auf das Web beschränkt bleiben. Es ist vorstellbar, dass die Daten künftig auch zur Erstellung eines gedruckten Vorlesungsverzeichnisses genutzt werden könnten.

Als Grundlage für die Umsetzung wurden vorhandene Daten aus einer bereits bestehenden MySQL-Datenbank zur Stundenplanerstellung verwendet.

**Schlagwörter:** Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), PHP, Auskunftssystem, Stundenplansystem, Hochschule der Medien Stuttgart

## Abstract

This thesis describes the setting up and the implementation of a calendar system based on XML. The development of this system is illustrated using the example of an online university calendar for the Department of Information and Communication at the School of Media, Stuttgart.

Compared with the more common usage of HTML to present university calendars on the web, the representation of the curriculum data with XML and its transformation with XSLT is more advantageous. This method enables the user to generate individually compiled output lists for various groups and persons.

Furthermore, the use of the XML representation of curriculum data need not be limited to online presentation. It provides the opportunity for the creation of outputs in other media, such as a printed university calendar.

The data processed in the system were taken from an existing MySQL data base that is currently used for the purpose of creating university calendars.

**Keywords:** Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), PHP, information system, calendar system, School of Media Stuttgart

# Inhaltsverzeichnis

<b>Kurzfassung .....</b>	<b>2</b>
<b>Abstract .....</b>	<b>3</b>
<b>Inhaltsverzeichnis .....</b>	<b>4</b>
<b>Abbildungsverzeichnis .....</b>	<b>7</b>
<b>Tabellenverzeichnis .....</b>	<b>7</b>
<b>Abkürzungsverzeichnis .....</b>	<b>8</b>
<b>Vorwort .....</b>	<b>10</b>
<b>1     Überblick .....</b>	<b>11</b>
1.1     Situation.....	11
1.2     Inhalt dieser Arbeit.....	12
<b>2     Ausgangspunkt und Ziele.....</b>	<b>14</b>
<b>3     Stand der Technik.....</b>	<b>16</b>
3.1     Bisherige Lösung im Fachbereich IuK der Hochschule der Medien Stuttgart.....	16
3.2     Vergleichbare, auf XML-Technologie basierende Lösungen.....	19
3.2.1     Technische Universität Berlin.....	19
3.2.2     Universität Magdeburg.....	21
3.3     Grenzen existierender Lösungen bei einem möglichen Einsatz in der HdM.....	23
<b>4     Anforderungen an ein Auskunftssystem für die Hochschule der Medien.....</b>	<b>27</b>
4.1     Grundlegende Funktionalitäten.....	27
4.2     Inhaltliche Anforderungen .....	28
<b>5     Werkzeuge und Technologien.....</b>	<b>30</b>
5.1     Datenübernahme und –repräsentation.....	30
5.1.1     Microsoft Access.....	30
5.1.2     Datenübernahme mit MyODBC.....	30
5.1.3     Extensible Markup Language (XML) .....	31
5.2     Datentransformation und –selektion .....	33
5.2.1     Cascading Style Sheets (CSS) .....	33

---

5.2.2	XSLT und XPath .....	34
5.2.3	XSLT-Prozessoren .....	35
5.3	Datenpräsentation im Web .....	39
5.3.1	PHP und Hypertext Markup Language (HTML).....	39
5.3.2	Cascading Style Sheets (CSS) .....	39
<b>6</b>	<b>Beschreibung der durchgeführten Arbeiten.....</b>	<b>41</b>
6.1	Analyse der Studien- und Prüfungsordnung.....	41
6.1.1	Studiengänge IM, ÖB und WB.....	42
6.1.2	Studiengang IW.....	43
6.1.3	Studiengänge BMB und BMD .....	43
6.1.4	Studiengang ID .....	44
6.1.5	Masterstudiengänge BM und IW.....	44
6.1.6	Zusammenfassung.....	45
6.2	Analyse der Stundenplandaten.....	46
6.2.1	Konfiguration und Import der Datenquelle .....	46
6.2.2	Analyse des Datenbestandes .....	48
6.3	XML-Dokumentstruktur .....	52
6.4	Erstellung von XML-Dateien mit Hilfe von Microsoft Access.....	57
6.5	Erstellung von XSLT-Stylesheets zur Transformation in HTML.....	64
6.5.1	Gestalterische Aspekte.....	64
6.5.2	Stundenplanauskunft.....	65
6.5.3	Anzeige fakultativer Lehrveranstaltungen.....	73
6.5.4	Dozentenauskunft .....	76
6.5.5	Raumauskunft.....	79
6.6	Aufruf der Veranstaltungspläne im Netz .....	79
6.6.1	Nutzerführung und Erfassung der Parameterwerte .....	80
6.6.2	Aufruf des XSLT-Prozessors.....	84
<b>7</b>	<b>Funktionsweise des Systems .....</b>	<b>86</b>
7.1	Funktionsweise aus der Sicht des Bearbeiters .....	86
7.1.1	Aktualisierung der XML-Repräsentanz des Stundenplans .....	86
7.1.2	Anpassung der Nutzersteuerung .....	87
7.2	Funktionsweise aus der Sicht des Benutzers.....	89
7.2.1	Stundenplanauskunft für Studierende.....	90
7.2.2	Dozentenauskunft .....	94
7.2.3	Raumauskunft.....	95
<b>8</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>97</b>

---

<b>Anhang A: CD-ROM .....</b>	<b>98</b>
<b>Literaturverzeichnis .....</b>	<b>100</b>
Printmedien .....	100
Onlinemedien .....	101
<b>Erklärung .....</b>	<b>103</b>

## Abbildungsverzeichnis

Abbildung 1: Ausgabe der derzeitigen Stundenplanauskunft des FB IuK.....	17
Abbildung 2: Ausgabe der derzeitigen Dozenten- und Raumauskunft des FB IuK .....	18
Abbildung 3: Die XML-Stundenplanauskunft der Universität Magdeburg.....	22
Abbildung 4: Vereinfachtes Verarbeitungsmodell von Cocoon.....	23
Abbildung 5: Vereinfachte Darstellung der Arbeitsweise eines XSLT-Prozessors .....	36
Abbildung 6: Konfiguration des MyODBC-Treibers .....	47
Abbildung 7: Auswahl der Datenquelle.....	47
Abbildung 8: Abfragen zur Gewinnung der Daten für die XML-Darstellung des Stundenplans .....	59
Abbildung 9: Formulierung von Abfragen mit Hilfe des Ausdrucks-Editors .....	61
Abbildung 10: Abfrage zur Gewinnung der Daten für die XML-Darstellung der Dozenten- und Raumauskunft .....	62
Abbildung 11: Struktureller Aufbau aller Stundenplananzeigen.....	64
Abbildung 12: Kopiervorgang mit WS_FTP.....	87
Abbildung 13: Eingangsseite des Auskunftssystems.....	90
Abbildung 14: Stundenplanauskunft: Anzeige der fakultativen Veranstaltungen .....	91
Abbildung 15: Stundenplanauskunft: Auswahl des Studiengangs .....	91
Abbildung 16: Stundenplanauskunft: Auswahl des Semesters und der Module .....	92
Abbildung 17: Stundenplanauskunft: Ausgabe (oberer Teil) .....	93
Abbildung 18: Stundenplanauskunft: Ausgabe (unterer Teil) .....	93
Abbildung 19: Dozenten- und Raumauskunft .....	94
Abbildung 20: Dozentenauskunft: Ausgabe .....	95
Abbildung 21: Raumauskunft: Ausgabe .....	96

## Tabellenverzeichnis

Tabelle 1: Überblick über die Wahlmöglichkeiten in den Studiengängen.....	45
Tabelle 2: Zur Generierung des Stundenplans erforderliche Parameter .....	80
Tabelle 3: Durchzuführende Anpassungen in der Stundenplanauskunft.....	89
Tabelle 4: Inhalt der CD-ROM .....	98

## Abkürzungsverzeichnis

BM	Bibliotheks- und Medienmanagement
BMB	Bibliotheks- und Medienmanagement (Bachelorstudiengang)
BMD	Bibliotheks- und Medienmanagement (Diplomstudiengang)
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
DTD	Document Type Definition
FTP	File Transfer Protocol
HBI	Hochschule für Bibliotheks- und Informationswesen
HdM	Hochschule der Medien
ID	Informationsdesign (Bachelorstudiengang)
IE5	Microsoft Internet Explorer 5
IM	Informationsmanagement (Diplomstudiengang)
IW	Informationswirtschaft (Diplomstudiengang)
IuK	Information und Kommunikation
HTML	Hypertext Markup Language
Master BM	Bibliotheks- und Medienmanagement (Masterstudiengang)
Master IW	Informationswirtschaft (Masterstudiengang)
MS	Microsoft
MSXML	Microsoft XML-Parser
ÖB	Öffentliche Bibliotheken (Diplomstudiengang)
ODBC	Open Database Connectivity



---

PDF	Portable Document Format
PHP	PHP Hypertext Preprocessor
SPO	Studien- und Prüfungsordnung
SWS	Semesterwochenstunden
vHdM	virtuelle Hochschule der Medien
WAP	Wireless Application Protocol
WB	Wissenschaftliche Bibliotheken (Diplomstudiengang)
WML	Wireless Markup Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

## **Vorwort**

Grundlage der in dieser Arbeit beschriebenen Systementwicklung ist die MySQL-Datenbank ws2001 auf dem Server vHdM. Änderungen an der Struktur der Datenbank können sich nach dem Abschluss der vorliegenden Arbeit negativ auf das Gesamtergebnis auswirken. Die verwendete Datenbank wurde während der Erstellung der Arbeit durch deren Administrator weiter modifiziert. Um negative Auswirkungen auf das Resultat dieser Arbeit zu vermeiden, bildet die Datenbankstruktur und der Datenbestand vom 24. September 2001 die Grundlage der Systembeschreibung.

Mein Dank für die Unterstützung bei der Erstellung dieser Arbeit gilt Frau Dorothee Schlör, für ihre Geduld bei zahlreichen Nachfragen in Bezug auf die Stundenplandatenbank sowie die Studien- und Prüfungsordnungen der Studiengänge.

Für die Hilfe bei der Installation und Konfiguration mehrerer XSLT-Prozessoren auf dem Server vHdM und für die hilfreichen Hinweise im Zusammenhang mit PHP danke ich Herrn David Prüm.

# 1 Überblick

## 1.1 Situation

Wie die meisten Universitäten und Fachhochschulen stellt inzwischen auch die Stuttgarter Hochschule der Medien ihr Vorlesungsverzeichnis online zur Verfügung. Bei der überwiegenden Anzahl dieser Vorlesungsverzeichnisse im Netz lässt sich feststellen, dass die Möglichkeiten der Informations- und Internettechnologie, beispielsweise das individuelle Abfragen von Datenbeständen, nur begrenzt genutzt werden. Meist handelt es sich um seitenlange tabellarischen Auflistungen von Vorlesungsterminen. Die Folge sind lange und unübersichtliche Dokumente, durch die sich die Nutzer kämpfen müssen.

An einigen Stellen, wie auch an der HdM, sind weitergehende Ansätze verwirklicht. Jedoch fehlt es bisher an einer Lösung, bei der die Nutzer die Möglichkeit haben, sich persönliche Stundenpläne ausgeben zu lassen.

In dieser Arbeit wird die Entwicklung eines Prototyps beschrieben, der auf der Basis von XML-Technologie die Generierung individueller Terminpläne<sup>1</sup> ermöglicht. Dieses Informationsmanagement-System beruht dabei auf der Transformation von Daten, die durch XML beschrieben werden. Im Verlauf dieser Transformation werden die Daten für eine Ausgabe im Web selektiert, sortiert und formatiert.

Der Einsatz dieses Systems wird am praktischen Beispiel des Fachbereichs Information und Kommunikation (IuK) der im September 2001 neu gegründeten Hochschule der Medien (HdM) Stuttgart gezeigt. Das System ermöglicht die Generierung eines persönlichen Stundenplans für jeden Studenten<sup>2</sup> unter Berücksichtigung dessen Studiengangs, Fachsemesters und seiner individuellen Wahlmodulen. Außerdem sind die Berücksichtigung fa-

---

<sup>1</sup> Nachfolgend sind unter „Terminplänen“ und „Veranstaltungsplänen“ Pläne sowohl für Studierende und Lehrkräfte, als auch Raumbelegungspläne zu verstehen. Der Begriff „Stundenpläne“ wird nur im Zusammenhang mit Studierenden, also als „Veranstaltungspläne für Studierende“, verwendet.

kultativer Lehrveranstaltungen, die Ausgabe von Wochenplänen für Professoren sowie von Raumbellegungsplänen vorgesehen.

Das derzeit noch im Einsatz befindliche System leistet nur einen Teil dieser Funktionalitäten. Eine personalisierte Ausgabe von Stundenplänen ist dort nicht möglich. Die in dieser Arbeit beschriebene Methode stellt also eine Verbesserung durch Erhöhung der Ausgabequalität dar.

## 1.2 Inhalt dieser Arbeit

Diese Arbeit umfasst die Darstellung der Entwicklung eines Auskunftssystems zur Erstellung individueller Terminpläne. Der Schwerpunkt liegt in der Beschreibung der verwendeten Technologien, der Darlegung der durchgeführten Arbeiten und der Erläuterung der Funktionsweise des Systems. In weiteren Abschnitten werden die Ausgangssituation, der Stand der Technik und die Anforderungen an das zu erstellende System beschrieben.

In Kapitel 2 werden die Ausgangssituation und die Ziele, die in dieser Arbeit erreicht werden sollen, dargestellt. Ferner wird hier der Benutzerkreis und deren Bedürfnisse vorgestellt.

Kapitel 3 zeigt die bisherige Vorgehensweise zur Stundenplanerstellung an der ehemaligen Hochschule für Bibliotheks- und Informationswesen (seit September 2001: Fachbereich IuK der Hochschule der Medien) auf. Darüber hinaus werden an dieser Stelle ähnliche Systeme vorgestellt, deren Funktionsweise kurz beschrieben, sowie deren Möglichkeiten für einen Einsatz im Fachbereich<sup>3</sup> ausgelotet.

Die Anforderungen, die an eine derartige Lösung im vorgegebenen Umfeld gestellt werden, werden in Kapitel 4 dargelegt. Dabei wird zwischen den geforderten Funktionalitäten eines solchen Systems und dem zu umfassenden Inhalt unterschieden.

---

<sup>2</sup> Zum Zwecke der besseren Lesbarkeit wird in der gesamten Arbeit nur die maskuline Form (z.B. Student, Benutzer, etc.) verwendet. Diese impliziert selbstverständlich jeweils auch die feminine Form.

<sup>3</sup> Im Folgenden ist unter dem Begriff „Fachbereich“ der Fachbereich Information und Kommunikation der Hochschule der Medien zu verstehen, da sich der Einsatz des beschriebenen Systems auf diesen beschränkt.

In Kapitel 5 werden die verwendeten Werkzeuge und Technologien, sowie zum Teil deren zur Verfügung stehenden Alternativen beschrieben. Zudem wird erörtert, warum die gewählte Technologie verwendet wurde.

Kapitel 6 beschäftigt sich konkret mit den zur Erstellung des Systems durchgeführten Arbeiten. Diese sind

- die Analyse der Stundenplandaten sowie der Studien- und Prüfungsordnungen,
- die Erstellung einer XML-Dokumentstruktur,
- die Übernahme der Daten aus der existierenden Stundenplandatenbank in ein entsprechendes XML-Dokument,
- die Erstellung von XSLT-Stylesheets<sup>4</sup> zur Transformation in HTML und
- der Aufruf des XSLT-Prozessors über Formulare im Web und die Ergebnisanzeige.

Die Funktionsweise des Systems aus den Sichtweisen des Benutzers und des Bearbeiters beschreibt das Kapitel 7.

Kapitel 8 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf die weiteren Entwicklungsmöglichkeiten.

Die Vorgehensweise bei der Installation von XSLT-Prozessoren auf dem Webserver, sowie das Verfügbarmachen und Konfigurieren dieser Tools auf dem Server sind nicht Gegenstand dieser Arbeit und werden daher nicht beschrieben. Ebenso ist der Prozess der Planung von Lehrveranstaltungen und damit zusammenhängender Fragestellungen aus hochschulorganisatorischer Sicht nicht Inhalt dieser Arbeit. Diese werden daher nur soweit behandelt, wie sie in einem Zusammenhang mit dem Thema dieser Arbeit stehen.

---

<sup>4</sup> Extensible Stylesheet Language Transformations: vgl. Kapitel 5.2.2

## 2 Ausgangspunkt und Ziele

Im Fachbereich IuK der Hochschule der Medien gibt es zur Zeit neun verschiedene Studiengänge. Einige sind völlig eigenständig, andere werden zum Teil gemeinsam mit verwandten Studiengängen durchgeführt. In allen Studiengängen müssen sich die Studenten durch verschiedene Wahlmodule spezialisieren. Die meisten Studiengänge sehen neben Wahlmodulen einen Pflichtteil vor, manche wiederum bestehen nur aus Wahlmodulen.

Die ganze Komplexität dieser Struktur wird in Kapitel 6 eingehender beschrieben. Bereits an dieser Stelle wird ersichtlich, dass es schwierig ist, in dieser Vielfalt den Überblick zu bewahren.

Ziel dieser Arbeit ist es, durch ein neues Auskunftssystem eine Erleichterung bei der Zusammenstellung des Stundenplans zu erreichen und die Auskunftsmöglichkeiten nach Vorlesungsterminen zu verbessern.

Das hier beschriebene System bildet neben der Verwendung im Web auch eine Grundlage für die Generierung weiterer Ausgaben in anderer Form, bzw. für andere Medien. Dabei ist in erster Linie an ein gedrucktes Vorlesungsverzeichnis oder die Konvertierung in PDF-Dateien<sup>5</sup> zu denken.

Als Benutzer der Entwicklung sind in erster Linie die Angehörigen der Hochschule der Medien, insbesondere die Mitglieder des Fachbereichs IuK vorgesehen. In einem erweiterten Kreis könnten dies auch Interessenten von außerhalb der Hochschule sein. Dabei ist im Wesentlichen an die große Zahl von Lehrbeauftragten zu denken, welchen die Möglichkeit gegeben werden sollte, das System auch ohne einen Zugang zum Intranet nutzen zu können. Da keine vertraulichen Informationen verarbeitet werden, die anderen Personen vor-  
enthalten werden müssten, scheint eine Beschränkung der Nutzungsmöglichkeiten scheint

---

<sup>5</sup> PDF (Portable Document Format) ist ein plattformübergreifendes Dokumentenformat der Firma Adobe, mit dem sich aus Texten und Grafiken bestehende Dokumente austauschen (erzeugen und darstellen) lassen.

daher in keinem Falle notwendig. Das System kann somit frei zugänglich auf vHdM<sup>6</sup>, dem „Lehre-Server“ des Fachbereichs, betrieben werden.

---

<sup>6</sup> URL des „Lehre-Servers“: <http://v.hdm-stuttgart.de>; URL des beschriebenen Prototyps auf dem Leheserver: <http://v.hdm-stuttgart.de/~spaleck>

## 3 Stand der Technik

Die Beschreibung des Stands der Technik findet aus zwei Perspektiven statt. Zum einen wird die bisherige Lösung und Vorgehensweise im Fachbereich geschildert, zum anderen werden vergleichbare, auf XML-Technologie beruhende Lösungen, die als Stundenplanauskunft dienen, vorgestellt. Im Anschluss werden die zuvor vorgestellten Systeme hinsichtlich ihrer Grenzen bei einem möglichen Einsatz in der Hochschule der Medien analysiert.

### 3.1 Bisherige Lösung im Fachbereich IuK der Hochschule der Medien Stuttgart

Die Stundenplanerstellung im Fachbereich<sup>7</sup> folgte in den vergangenen Semestern keinem einheitlichen Schema. So wurde vom Entwickler der Stundenplanauskunft, Herrn Prof. Klaus Sailer, der auch für die technische Seite der Stundenplandatenbank zuständig ist, für jedes Semester eine neue Datenbank erstellt. Diese wies teilweise unterschiedliche Tabellen und in den Tabellen unterschiedliche Felder bzw. Feldbezeichnungen auf.

Ursache dessen mag die Einführung neuer, anders strukturierter Studiengänge gewesen sein, die diese Änderungen notwendig machten. Zweifellos wird die Weiterverarbeitung der Daten durch die Veränderung der Datenbankstruktur erheblich erschwert.

Die Stundenplanauskunft<sup>8</sup> baut auf diese Datenbank, die MySQL-Datenbank ws2001, auf. Von einer zentralen Seite im Netz sind dort die Stundenpläne der einzelnen Studiengänge und deren Fachsemester, sowie die Dozenten- und Raumauskunft abrufbar. Realisiert sind diese mit PHP<sup>9</sup>, mit dessen Hilfe Datenbankabfragen in der Stundenplandatenbank formuliert werden und deren Resultat als HTML-Seite ausgegeben wird. Die ausgewählten Werte (z.B. Studiengang und Fachsemester) werden von der Auswahlseite als Parameter übergeben.

---

<sup>7</sup> Bis Ende August 2001 noch Hochschule für Bibliotheks- und Informationswesen (HBI).

<sup>8</sup> Die Stundenplanauskunft ist im Internet unter <http://v.bdm-stuttgart.de/stundenplan.html> erreichbar.

<sup>9</sup> PHP Hypertext Preprocessor: serverseitige Skriptsprache



Abbildung 1 zeigt eine Ausgabe der Stundenplanauskunft am Beispiel des Studiengangs Informationsmanagement (7. Semester).

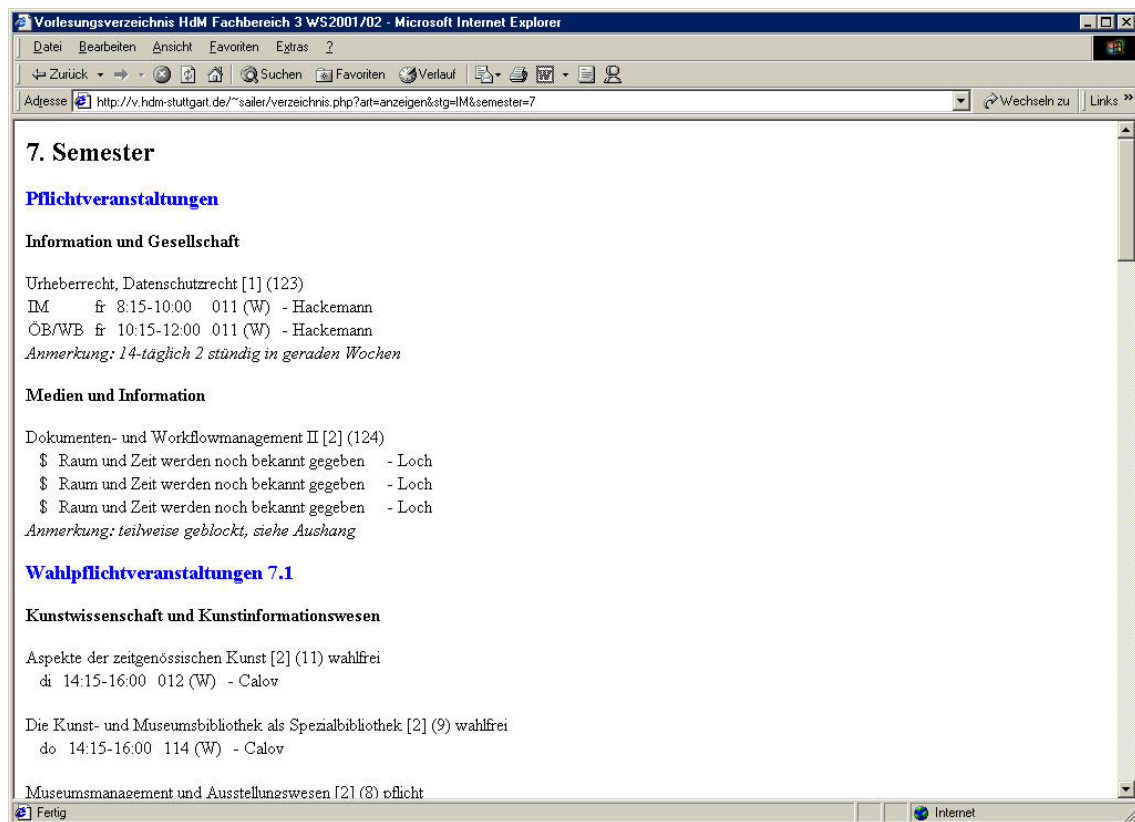


Abbildung 1: Ausgabe der derzeitigen Stundenplanauskunft des FB IuK<sup>10</sup>

Die Dozenten- und Raumauskunft gibt einen tabellarischen Wochenplan für die gewählte Kalenderwoche aus. Der Benutzer muss auf der Startseite der Dozenten- und Raumauskunft eine Kalenderwoche und entweder einen Dozenten oder einen Raum auswählen. Im oberen Frame werden zu den Terminen die betreffenden Raumnummern angezeigt (Abbildung 2 verdeutlicht die Dozenten- und Raumauskunft). Bei Auswahl eines Termins erscheinen im unteren Frame die zugehörigen Termindaten.

<sup>10</sup> HdM [2001c]

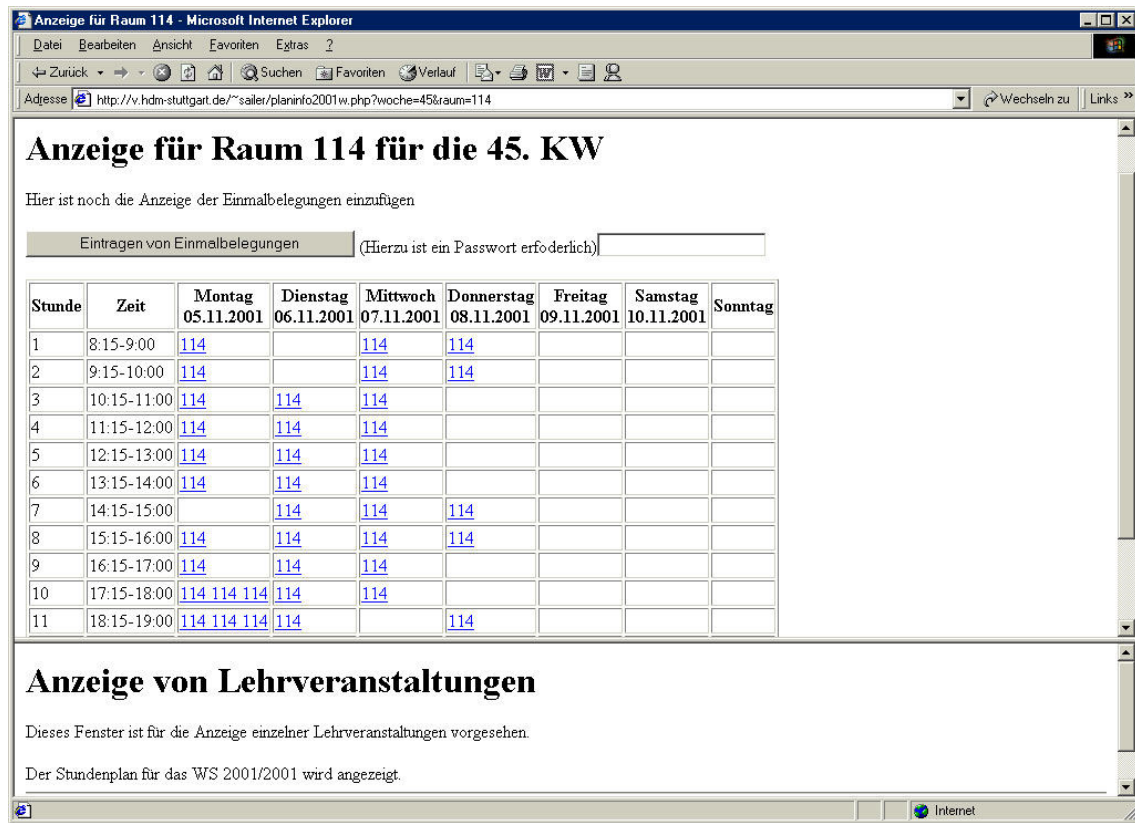


Abbildung 2: Ausgabe der derzeitigen Dozenten- und Raumauskunft des FB IuK<sup>11</sup>

Neben der Ausgabe der Termine bietet die Dozenten- und Raumauskunft auch die Möglichkeit einzelne Termine einzutragen, d.h. Räume, für sich nicht periodisch wiederholende Lehrveranstaltungen, zu reservieren (Einmalbelegung). Möglich ist dies nur mit einem gültigen Passwort über die Schaltfläche „*Eintragen von Einmalbelegungen*“. Die Daten der Einmalbelegungen werden in einer eigenen Datenbank abgespeichert und verwaltet. Es handelt sich hier also nicht nur um ein Auskunftssystem, sondern auch um ein Raumplansystem.

Zeitgleich zu der hier vorliegenden Arbeit wurde im Fachbereich IuK von Prof. Klaus Sailer ebenfalls an einer Stundenplanauskunft auf XML-Basis gearbeitet. Diese war zum Zeitpunkt des Abschlusses dieser Arbeit jedoch noch nicht soweit fortgeschritten, dass eine eingehende Darstellung an dieser Stelle möglich bzw. sinnvoll gewesen wäre. Daher sollen im Folgenden nur deren Grundzüge vorgestellt werden, die auf dem Stand von Anfang Oktober 2001 beruhen.

<sup>11</sup> HdM [2001d]

Die Darstellung des Stundenplans durch XML wurde mit Hilfe der oben beschriebenen Stundenplanauskunft ermöglicht. Dabei werden die Stundenplandaten durch eine XML-Datei repräsentiert und durch ein PHP-Skript für die Ausgabe im Web verarbeitet. Die Formatierung der dargestellten Inhalte findet durch Cascading Style Sheets (CSS) statt. Eine Transformation der XML-Datei, die eine Selektion einzelner Veranstaltungen oder Sortierungen vornimmt ist derzeit noch nicht vorhanden. Wie CSS zur Formatierung von XML-Dokumenten verwendet werden können, wird in Abschnitt 5.2.1 dargestellt.

## **3.2 Vergleichbare, auf XML-Technologie basierende Lösungen**

Die Recherche nach vergleichbaren Systemen, die ebenfalls auf XML-Technologie beruhen, ergab, dass es bislang kaum Systeme in diesem Einsatzbereich gibt. Die Mehrzahl der Lösungen entstand als Prototyp in Programmierübungen an Universitäten und Fachhochschulen und waren zum Teil zugangsgeschützt oder nicht mehr erreichbar. Die nachstehenden Abschnitte beschreiben die Systeme, die noch zugänglich sind bzw. einen interessanten Ansatz im Rahmen dieser Arbeit verfolgen.

### **3.2.1 Technische Universität Berlin**

Im Fachbereich Informatik der Technischen Universität Berlin fand im Sommersemester 2000 ein Projektseminar statt, das sich zum Ziel gesetzt hatte, ein web-basiertes Informationssystem für Lehrveranstaltungen zu erstellen. Die Seminarteilnehmer verfolgten unterschiedliche Ansätze zur Umsetzung. Die Ergebnisse des Seminars sind heute nur noch in geringem Maße im Internet verfügbar, der XML-Ansatz leider nicht mehr. Da es sich allerdings um einen sehr interessanten Entwurf handelt, soll dessen theoretisches Konzept hier dennoch kurz vorgestellt werden.

Stede/Koch [2000] führen aus, der erste wichtige Bestandteil zur Benutzung des Systems sei das Nutzerprofil. Dieses enthalte die Veranstaltungen, die der Student bereits besucht und bestanden habe sowie dessen bevorzugte Themengebiete. Die Anzeige der Vorlesungstermine erfolge nach den Präferenzen des Studenten, wobei auch Termine angezeigt würden, die von der Person noch nicht belegt werden könnten. Die Ausgabe auf dem Bildschirm sei in drei Frames unterteilt: *Views*, *Courses* und *Schedule*.

In *Views* könne der Benutzer die Einstellungen zur Darstellung und Auswahl der angezeigten Termine treffen. In *Courses* würden die Details zu den Veranstaltungen dargestellt. An dieser Stelle könne der Termin auch ausgewählt, vorgemerkt oder fallengelassen werden. *Schedule* zeige tabellarisch den wöchentlichen Stundenplan an.

Das System überprüfe zudem die Auswahl des Benutzers. So werde dieser alarmiert, wenn

- er mehrere Veranstaltungen wähle, die sich zeitlich überschneiden,
- er zeitlich aufeinander folgende Veranstaltungen wähle, die an entfernten Orten stattfänden,
- er eine Veranstaltung wähle, die er bereits in vergangenen Semestern belegt und bestanden habe,
- er einen Termin wähle den er auf Grund der Teilnahmebedingungen noch nicht wählen dürfe.

Das oben beschriebene System ermöglicht den Studenten, ihre Kurse frei auszuwählen und gibt ihnen dabei größte Freiheit. Voraussetzung zur Benutzung ist demnach ein vollständiges Nutzerprofil mit allen Daten über bereits besuchte und erfolgreich abgeschlossene Veranstaltungen.

Die technische Implementierung des beschriebenen Prototyps an der TU Berlin stellen Stede/Koch [2000] folgendermaßen dar:

*„The standard ASCII list of courses offered in a term (as created by the department) is semi-automatically converted to an XML document. The different views of the course listings are generated via XSL-T with the Microsoft XML Parser MSXML2, which is integrated in Internet Explorer 5 (IE5). Given the current state of HTML technology, the constraints imposed by the pre-standard XSL portions in IE5 need to be respected, and the target language is currently HTML. [...] The dynamic (session-specific) user profile is stored as cookies.“*

Die XML-Datei wird hier aus einem bereits vorliegenden Datenbestand erzeugt. Ähnliche Voraussetzungen treffen auch auf die HdM zu, wo die Daten in einer MySQL-Datenbank

vorliegen. Zur Transformation verwendet das System XSLT mit MSXML2 als Parser<sup>12</sup>. MSXML2<sup>13</sup> ist ein XSLT-Prozessor, der die Transformation clientseitig vornimmt. Die Voraussetzung, um dieses System anwenden zu können, ist, dass der Anwender den MSXML-Parser auf seinem Rechner installiert haben muss. Die Abschnitte 5.2.2 (XSLT und XPath) und 5.2.3 (XSLT-Prozessoren) beschäftigen sich näher mit den technischen Spezifikationen dieser Technologien.

### 3.2.2 Universität Magdeburg

Die Otto-von-Guericke-Universität Magdeburg setzt seit dem Sommersemester 2001 ein XML-basiertes Stundenplanauskunftssystem ein. Dieses System ermöglicht die Ausgabe von Stundenplänen für Studenten, Veranstaltungsplänen für Mitarbeiter und Raumbelegungsübersichten. Des weiteren ist über das System eine Raumreservierung möglich. Vor der Ausgabe der Pläne erhält der Benutzer ein Auswahlformular, bei dem er entweder den Studiengang mit Jahrgang (Studenten), den Namen der Lehrkraft (Mitarbeiter) oder die Raumbezeichnung (Raumbelegung) auswählen kann. Abbildung 3 zeigt eine beispielhafte Ausgabe für einen Studiengang.

Die Ausgabe des Stundenplans enthält die wichtigsten Angaben zur Lehrveranstaltung. Sie erfolgt tabellarisch und chronologisch geordnet. Die Angabe des Typs gibt Aufschluss darüber, ob es sich um eine Vorlesung oder Übung handelt. Die Anzeige der Veranstaltungspläne der Mitarbeiter und der Raumbelegungspläne sind ähnlich aufgebaut.

Die Stundenplanauskunft der Universität Magdeburg wurde mit Hilfe von Cocoon<sup>14</sup> realisiert. Cocoon ist ein Web Publishing Framework und wurde im Rahmen des Apache XML-Projekts als Java-Servlet<sup>15</sup> entwickelt.

---

<sup>12</sup> Ein XML-Parser teilt ein XML-Dokument in dessen verschiedene Einheiten (Elemente, Attribute, etc.) auf und übergibt sie an eine Anwendung, z.B. einen XSLT-Prozessor. Man unterscheidet validierende und nicht-validierende Parser. Ein nicht-validierender Parser überprüft das Dokument auf Wohlgeformtheit, also auf Übereinstimmung mit der XML-Syntax. Ein validierender Parser kontrolliert zudem die Kongruenz des Dokuments mit dessen DTD. [vgl. Harold/Means 2001]

<sup>13</sup> aktuelle Version: MSXML3. Download über <http://msdn.microsoft.com/downloads>

<sup>14</sup> <http://xml.apache.org/cocoon/>

<sup>15</sup> Servlets sind meist kleine in der Programmiersprache Java geschriebene Programme. Im Gegensatz zu Java-Applets werden diese aber nicht auf dem Client, sondern auf dem Server ausgeführt.

**Stundenplan für BW0100**

Titel	Tag	Zeitraum	Typ	Raum
EDV	Montag	9:00 - 11:00	U	G02-113/RTL3
Stat. I	Montag	9:00 - 11:00	V	G16-H5 (478)
EDV	Montag	11:00 - 13:00	U	G02-113/RTL3
Bürgerl. Recht	Montag	15:00 - 17:00	V	G16-H5 (478)
Bürgerl. Recht	Montag	17:00 - 19:00	U	G16-H5 (478)
VWL A	Dienstag	9:00 - 11:00	U	G10-401 (60)
Statistik A	Dienstag	11:00 - 13:00	U	G10-111 (72)
Wirt.-pol.	Dienstag	15:00 - 17:00	U	G10-460 (60)
Mathematik II	Dienstag	17:00 - 19:00	U	G22A-020 (70)
Stat. I	Mittwoch	7:00 - 9:00	U	G16-H5 (478)
Entsch.-Th.	Mittwoch	9:00 - 11:00	V	G16-H5 (478); G22A-021 (148);
Mikroök.	Mittwoch	11:00 - 13:00	V	G16-H5 (478); G22A-021 (148);
Entsch. Th.	Mittwoch	15:00 - 17:00	U	G10-111 (72)
Öffentl. Recht	Mittwoch	17:00 - 19:00	V	G16-H5 (478)
EDV	Donnerstag	9:00 - 11:00	U	G02-113/RTL3
Stat. I	Donnerstag	11:00 - 13:00	V	G16-H5 (478)
Mikroök.	Donnerstag	15:00 - 17:00	U	G16-H5 (478); G22A-021 (148);
Öffentl. Recht	Donnerstag	17:00 - 19:00	V	G16-H5 (478)

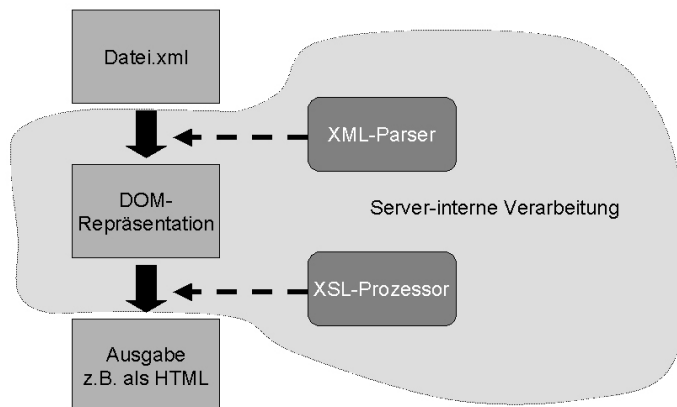
Abbildung 3: Die XML-Stundenplanauskunft der Universität Magdeburg<sup>16</sup>

Um Cocoon verwenden zu können, muss es also in den Web oder Anwendungsserver integriert werden. Der gesamte Verarbeitungsprozess findet demnach auf dem Server statt. Für den Anwender ist dies von großem Vorteil, da er beispielsweise keinen XSLT-Prozessor auf seinem Rechner installiert haben muss.

Die Abbildung 4 verdeutlicht die Arbeitsweise von Cocoon. Ein Parser wertet das XML-Dokument auf dem Server aus und bringt es in eine DOM-Repräsentation<sup>17</sup>. Der XSLT-Prozessor verwendet wiederum die DOM-Repräsentation um daraus die Ausgabe zu generieren.

<sup>16</sup> Universität Magdeburg [2001]

<sup>17</sup> DOM (Document Object Model) ist eine plattform- und sprachenunabhängige Schnittstelle die es ermöglicht auf die Struktur und den Inhalt von XML- und HTML-Dokumenten zuzugreifen und diese zu manipulieren. DOM Level 2 ist bereits eine W3C Recommendation, DOM Level 3 liegt derzeit als Working Draft beim W3C vor. [vgl. Le Hégarret/Wood 2001]

Abbildung 4: Vereinfachtes Verarbeitungsmodell von Cocoon<sup>18</sup>

### 3.3 Grenzen existierender Lösungen bei einem möglichen Einsatz in der Hochschule der Medien

Die oben beschriebenen Stundenplansysteme können nicht unmittelbar auf die Gegebenheiten der HdM übertragen werden. In diesem Abschnitt werden sie hinsichtlich eines Einsatzes im Fachbereich IuK bewertet. Auch die derzeitige Stundenplanauskunft am Fachbereich hat in ihren Anwendungsmöglichkeiten Grenzen. Bei dem im Abschnitt 3.1 beschriebenen XML-Stundenplansystem, das zur Zeit am Fachbereich entwickelt wird, handelt es sich um einen noch in der Entwicklungsphase befindlichen Prototyp. Dieser wird in der folgenden Ausführung daher außer Acht gelassen.

Das bisher im Fachbereich eingesetzte Stundenplan-Auskunftssystem hat sich während der vergangenen Semester durchaus bewährt. Bei genauerer Analyse fallen jedoch dessen Nachteile und Grenzen auf. Zum Beispiel ist eine nach dem individuellen Bedürfnissen der Benutzer ausgerichtete Ausgabe nicht möglich. Auch werden die Wahlmodule der Studenten bei der Generierung der Anzeige nicht berücksichtigt. Das bedeutet, ein Student erhält in der Ausgabe alle, also auch nicht relevante, Lehrveranstaltungen für seinen Studiengang und Studiensemester ausgegeben. Darin sind sämtliche für die Studierenden dieses Studiengangs und Semesters in Wahlmodulen wählbaren Veranstaltungen enthalten. Idealer-

---

<sup>18</sup> nach Mintert/Menge [2000]

weise sollten nur die Termine angezeigt werden, die für den betreffenden Studenten auch zu belegen sind.

Der zweite offensichtliche Nachteil der momentanen Lösung an der HdM ist die zeitlich unsortierte Ausgabe der Termine. Die Ausgabe erfolgt nach Studiengang und Modul geordnet, d.h. unter der Überschrift eines Moduls werden sämtliche in diesem Bereich angebotenen Lehrveranstaltungen aufgeführt.

Ein zusätzliches Manko der Ausgabe ist die zum Teil unzureichende Beschriftung der Ausgabe. Am Beispiel des ersten in Abbildung 1 dargestellten Termins wird dies deutlich. Der Benutzer erfährt mehrere Zahlen, deren Bedeutung bleibt jedoch offen und ist zumindest für Studienanfänger schwer nachzuvollziehen.

Im Hinblick auf die Darstellung bietet das Auskunftssystem keine Vorteile gegenüber dem gedruckten Vorlesungsverzeichnis der Hochschule da die Darstellungsweise der Termine im Auskunftssystem bis auf die Angabe des Sigels<sup>19</sup> völlig identisch ist.

Dagegen zeichnet sich der an der TU Berlin implementierte Prototyp durch seine umfangreichen und vielseitigen Funktionalitäten aus. Es handelt sich dort um eine in hohem Maße an die Situation des Nutzers anpassungsfähige Lösung. Der Einsatz eines in diesem Umfang flexiblen Systems ist am Fachbereich IuK nicht erforderlich.

Wegen der Struktur der Studiengänge am Fachbereich, die neben Pflichtfächern, einen in nicht unerheblichem Umfang nach den gewählten Wahlmodulen ausgerichteten Stundenplan vorsehen [vgl. HBI 1997, HdM 2001a, HdM 2001b], ist es nicht notwendig alle Vorlesungen anzuzeigen. Die Wählbarkeit der Vorlesungen richtet sich nur sehr bedingt nach den Präferenzen der Studenten, sondern vielmehr nach deren Wahlmodulen<sup>20</sup>. Die zeitliche Einplanung der Veranstaltungen findet bereits in der Phase der Planung der Veranstaltungen unter Berücksichtigung der starren Stundenplanstruktur statt. Vom Stundenplanamt werden festgelegte Zeitfenster für bestimmte Wahlbereiche und Pflichtbereiche vorreserviert. Diese Maßnahme schlägt sich später in der geringen Zahl der Überschneidungen nieder.

---

<sup>19</sup> Die Bedeutung des Sigels wird in Kapitel 6 eingehend behandelt.

<sup>20</sup> Die Festlegung auf bestimmte Wahlmodule drückt gewissermaßen die Präferenz des Studierenden aus.



Eine Kontrolle, ob auf eine Veranstaltung nachfolgende Termine an entfernten Orten stattfinden, ist bislang ebenfalls nicht nötig, da durch Unterbringung des Fachbereichs an einem Standort weite Wege vermieden werden. Größere Distanzen würden sich erst ergeben, sobald Lehrveranstaltungen in größerer Zahl mit anderen Fachbereichen der Hochschule durchgeführt würden.

Eine Prüfung auf noch nicht erbrachte Vorleistungen für eine Lehrveranstaltung ist auf Grund der starren Strukturen der Studienordnung ebenfalls nicht zwingend notwendig.

Ein Problem technischer Art beim Einsatz im Fachbereich IuK stellt die Verwendung des Parsers MSXML2 dar. Dieser wird clientseitig ausgeführt und ist in den IE5 integriert. Zum gegenwärtigen Zeitpunkt wird im Fachbereich noch Netscape Navigator als Standard-Browser verwendet, der Internet Explorer ist dort nur auf wenigen Rechnern installiert. Netscape Navigator ist bislang jedoch nicht in ausreichendem Umfang fähig XML-Dateien komfortabel zu verarbeiten. Eine Umsetzung dieser Lösung im Umfeld der HdM ist daher nicht ohne weiteres möglich.

Die Stundenplanauskunft der Universität Magdeburg behebt die Mängel des Stundenplansystems des Fachbereichs zum Teil. Die Darstellung ist chronologisch geordnet und alle wichtigen Informationen für die Ausgabe sind vorhanden. Bei einem Einsatz in der HdM wären jedoch noch weitere Informationen wie zum Beispiel das Modul, die Art der Veranstaltung oder weitere Bemerkungen notwendig. Außerdem würde es bei einem Einsatz im Fachbereich nicht ausreichen, nur nach dem Studiengang und dem Jahrgang zu selektieren. Eine weitere Einschränkung der Ausgabe nach den Modulen wäre dann unbedingt erforderlich.

Der bedeutendere Nachteil wäre in diesem Fall jedoch die Festlegung auf Cocoon, eine Lösung die umfangreiche und komplizierte Installations- und Konfigurationsarbeiten voraussetzen würde [vgl. McLaughlin 2001, S. 239ff]. Mintert und Menge [2000] weisen darauf hin, Cocoon unterstütze als Parser derzeit lediglich Apache Xerces und Sun ProjectX. Des weiteren stünden als XSLT-Prozessoren nur Apache Xalan und XT zur Verfügung. Man müsse sich also neben der erfolgreichen Installation von Cocoon auch um die Installationen des XSLT-Prozessors und Parsers kümmern.

---

Die Verwendung eines derart großen Systems wie Cocoon stellt deshalb aus der Sicht des Autors einen unverhältnismäßig großen Aufwand bei der Entwicklung des dargestellten Prototyps dar.

## **4 Anforderungen an ein Auskunftssystem für die Hochschule der Medien**

An das Auskunftssystem, das zur Information über Lehrveranstaltungen dienen soll, werden von verschiedener Seite unterschiedliche Anforderungen gestellt. Auf der Seite der Nutzer des Systems müssen zum einen die Anforderungen der Studenten berücksichtigt werden, zum anderen die Bedürfnisse der Professoren und der Verwaltung. Es sind aber auch die Anliegen derjenigen zu beachten, die das System künftig administrativ betreuen werden.

Nachfolgend werden diese Bedürfnisse, die sich in Anforderungen funktioneller Art und inhaltlicher Art unterscheiden lassen, beschrieben.

### **4.1 Grundlegende Funktionalitäten**

Um den Anforderungen aller Nutzer und der Bearbeiter gerecht zu werden sollte das Auskunftssystem die folgenden grundlegenden Funktionalitäten bereitstellen:

- Das System soll für den Benutzer leicht zu bedienen sein und sich möglichst selbst erklären.
- Die Anpassungsarbeiten, d.h. die Angleichung an die Gegebenheiten eines neuen Semesters, sollen vom Bearbeiter möglichst einfach durchgeführt werden können.
- Es sollen Ausgaben für Studenten, Dozenten (Professoren und Lehrbeauftragte) und Räume, in denen Lehrveranstaltungen stattfinden, möglich sein.
- Eine Übersicht über alle angebotenen fakultativen Lehrveranstaltungen soll abrufbar sein.
- Die Ausgabe für Studenten soll personalisierbar sein, d.h. es sollen nur solche Termine ausgegeben werden, die für den Studenten in seinem Studiengang und Fachsemester als Pflichtveranstaltung vorgeschrieben sind und solche, die er durch die Belegung seiner individuellen Wahlmodule belegen muss bzw. kann.

- Die Ausgabe für Dozenten soll für jede Lehrkraft personalisierbar sein und alle vom jeweiligen Dozenten durchzuführenden Veranstaltungen umfassen.
- Die Ausgabe der Raumbellegungspläne soll alle im betreffenden Raum stattfindende Termine beinhalten.
- Die Veranstaltungspläne sollen alle chronologisch nach Tag und Uhrzeit sortiert sein.

## 4.2 Inhaltliche Anforderungen

Bei der Ausgabe der Termine sind zahlreiche Daten von Bedeutung. Einige Daten werden für alle, andere nur für einen Teil der Ausgaben benötigt:

- Die Angabe des **Wochentags**, der **Startzeit** und der **Endzeit** ist für die zeitliche Einordnung wichtig und muss daher bei allen Ausgaben vorgesehen werden.
- Die Angabe der **Art** der Lehrveranstaltung, also ob diese ein Pflicht- oder Wahlfach ist, ist nur für die Studenten von Bedeutung.
- Die Ausgabe des **Moduls** ist wichtig für die Ausgabe des Stundenplans der Studenten. Im Rahmen der Wahlmodule ist es ein Selektionskriterium und daher unbedingt vorzusehen. Auch für die Ausgabe der Dozenten- und Raumpläne ist die Angabe des Moduls eine wichtige Information, da des öfteren als Veranstaltungsname auch Bezeichnungen wie „Seminar“ oder „Übung“ vorkommen.
- Der **Name** der Veranstaltung ist unverzichtbar für dessen Identifikation durch den Nutzer. Das **Sigel** ist bei der Erstellung des Stundenplans und der Planung der Vorlesungszeiten und Räume durch das Stundenplanamt von Bedeutung. Eine Angabe ist daher durchaus sinnvoll und in Erwägung zu ziehen.<sup>21</sup>
- Manche Veranstaltungen werden unter dem gleichen Namen mehrfach aufgeführt, da sie zum Teil für unterschiedliche Studentengruppen zu verschiedenen Zeiten angeboten werden. Die Angabe der **Gruppe** ist somit sowohl für die Studenten und Lehrkräfte, als auch bei der Raumauskunft eine wichtige Information.

---

<sup>21</sup> Der Unterschied zwischen Name und Sigel der Veranstaltung wird in Kapitel 6 behandelt.

- **Anmerkungen** zum Termin sind von elementarer Bedeutung. Diese enthalten beispielsweise Informationen über die Durchführung (z.B. „Termin 14-täglich“), den Ort (z.B. „erster Termin in der Hochschule, danach an der VHS“), Verlegungen (z.B. „vorgezogenen Lehrveranstaltung aus dem 2. Semester“), benötigte Vorkenntnisse, die Art der Gruppeneinteilung, etc. Anmerkungen sind demnach bei allen Ausgabeformen vorzusehen.
- Der **Dozent** ist bei allen Ausgaben eine wichtige Information. Bei der Dozentenauskunft ist die Dozentenangabe Selektionskriterium und muss nicht in der Auflistung der Termine genannt werden.
- Der **Studiengang** und das **Semester** sind Auswahlkriterien für die Ausgabe der Stundenpläne der Studenten. Diese sind für die Generierung der Ausgabe wichtig, müssen bei der Auflistung der Lehrveranstaltungen jedoch nicht aufgeführt werden. Die Angabe von Studiengang und Semester ist in der Dozenten- und Raumauskunft nicht notwendig, kann aber eine nützliche Information darstellen. Hier tritt die Schwierigkeit auf, dass ein Termin für mehrere Studiengänge und/oder Semester durchgeführt wird und somit nicht eindeutig einem Studiengang und Semester zugeordnet werden kann.
- Die Angabe des **Raums** wird bei der Raumauskunft als Selektionskriterium benötigt. Bei der Stundenplan- und Dozentenauskunft muss der Raum angegeben werden um die Zuordnung der Termine auf die Räume darstellen zu können.
- Die Angabe der anrechenbaren **Semesterwochenstunden** (SWS) ist für Studenten eine wichtige Information bei der Zusammenstellung ihrer Stundenpläne. Da das System die Zusammenstellung des Stundenplans ermöglichen soll ist die Angabe der Semesterwochenstunden nötig. Für die Dozenten sind die SWS ohne Bedeutung, für die Raumauskunft ist lediglich die Durchführungsdauer von Belang.

## 5 Werkzeuge und Technologien

Zur Realisierung des Terminauskunftssystems wurde eine Vielzahl von Technologien verwendet. Dieses Kapitel beschreibt die **Auswahl** der technischen Hilfsmittel und Technologien sowie zum Teil deren **Funktionsweise**. Die konkrete **Anwendung** dieser Technologien im Rahmen des erstellten Prototyps sind Gegenstand von Kapitel 6. In den folgenden Abschnitten kommen die Technologien und Werkzeuge, die zur Datenübernahme aus der Quelldatenbank verwendet und zur Repräsentation der Daten angewendet wurden, zur Sprache. Die Datentransformation und schließlich die Darstellung im WWW bilden weitere Bereiche.

### 5.1 Datenübernahme und –repräsentation

Die für das hier entwickelte System benötigten Daten lagen bereits in einer vom Stundenplanamt des Fachbereichs verwendeten Datenbank vor. Diese wird auch von der alten Stundenplanauskunft verwendet und liegt auf vHdM, dem Lehre-Server des Fachbereichs, auf. Sie wurde mit dem Datenbanksystem MySQL erstellt.

Das Ziel in diesem ersten Schritt ist die Erzeugung einer XML-Datei zur Repräsentation aller für das Terminauskunftssystem erforderlichen Daten.

#### 5.1.1 Microsoft Access

Zur Erzeugung der XML-Darstellung des Datenbestandes wurde Microsoft Access verwendet. Es ist das im Fachbereich am weitesten verbreitete Datenbanksystem. Im Stundenplanamt wird es bereits zur Erstellung des Stundenplans eingesetzt, so dass die Mitarbeiter dort schon weitreichende Erfahrung mit diesem Produkt haben. MS Access verfügt darüber hinaus im Gegensatz zu MySQL, dem Datenbanksystem der Quelldatenbank, über eine komfortable graphische Benutzeroberfläche.

#### 5.1.2 Datenübernahme mit MyODBC

Das MySQL Datenbankhandbuch beschreibt MyODBC folgendermaßen:

*„MyODBC ist ein [...] ODBC-Treiber für Microsoft Windows Clients, die insbesondere speziell für MySQL angepasst wurden. Somit ist es möglich, mit Anwendungsprogrammen, wie Winword, Excel, Access, Staroffice [...] auf die MySQL Datenbank zuzugreifen.“ [Stepken 1999]*

Mit Hilfe von MyODBC kann also auf eine einfache Weise auf die MySQL-Datenbank zugegriffen und diese in MS Access importiert werden. Die Verwendung von MyODBC bietet sich auch deshalb an, da es kostenlos aus dem Internet heruntergeladen werden kann<sup>22</sup>.

MyODBC wird in der beschriebenen Entwicklung dazu benötigt um die auf dem Server vHdM in MySQL vorliegende Datenbank auf den Rechner des Benutzers zu laden um diese schließlich in Microsoft Access importieren zu können.

### 5.1.3 Extensible Markup Language (XML)

Um Text, oder ganz allgemein Daten, in Netzen darstellen zu können gibt es eine Vielzahl von Technologien. Am verbreitetsten, und noch immer so gut wie unverzichtbar ist dabei die Auszeichnungssprache Hypertext Markup Language (HTML). Neben HTML gibt es zahlreiche weitere Sprachen und Technologien. Eine dieser Sprachen ist die Extensible Markup Language<sup>23</sup> (XML), deren Syntax 1998 erstmals vom World Wide Web Consortium (W3C) vorgestellt wurde und seitdem eine enorme Aufmerksamkeit erfährt.

Eine Vielzahl von Gründen spricht dafür, dass die Bedeutung von XML in Zukunft auch weiter zunehmen wird:

- Durch die Entwicklung der Wireless Markup Language (WML), wurde eine XML-Anwendung geschaffen, die zur Darstellung von WAP-Seiten<sup>24</sup> auf Mobiltelefonen die führende Technologie auf diesem Gebiet bereitstellt.

---

<sup>22</sup> <http://www.mysql.org>

<sup>23</sup> In der Fachliteratur wird für XML sowohl die Langform *Extensible Markup Language* [z.B. Harold/Means 2001, Bach 2000], als auch die Form *Extended Markup Language* [z.B. Born 2001] verwendet. Diese Arbeit richtet sich nach der W3C Recommendation vom 6. Oktober 2000 [Bray et. al. 2000] sowie der Mehrzahl der Publikationen zum Thema XML und verwendet daher die Form *Extensible Markup Language*.

<sup>24</sup> WAP steht für Wireless Application Protocol

- XML wird zunehmend als Austauschformat zwischen verschiedenen Informationsservern und Datenbanksystemen verwendet, z.B. durch das Datenbanksystem *Tamino* der Software AG [vgl. Ziegler 2001].
- Die Plattformunabhängigkeit und die Unterstützung einer Vielzahl von Medien eröffnet unzählige Möglichkeiten zur Anwendung und Ausgabe von XML-Dokumenten [vgl. Born 2001].
- Laut Bach [2000] beziehen alle bedeutenden Unternehmen der IT- und Softwarebranche eine eindeutige Position zu XML, die der Technologie eine immense Bedeutung für die aktuellen und zukünftigen Entwicklungen in der Internetbranche im Speziellen sowie der IT-Branche im Allgemeinen zubilligen.

Auch die enorm steigende Zahl der Publikationen zum Thema XML bestätigt das zunehmende Interesse an dieser Technologie. Neben der Tatsache, dass sich XML immer mehr als Technologie für Internetprojekte durchsetzt bietet XML auch zahlreiche praktische Vorzüge im Zusammenhang mit dieser Arbeit.

XML ist eine sehr vielseitige Auszeichnungssprache. Harold und Means [2001] beschreiben deren Grundzüge folgendermaßen:

*„It defines a generic syntax used to mark up data with simple humanreadable tags. It provides a standard format for computer documents. [...] You can write your own programs that interact with, message, and manipulate data in XML documents.“*

Die Struktur der XML-Datei, die Document Type Definition (DTD) also, kann demnach im Gegensatz zu HTML selbst bestimmt werden. Aber nicht nur die DTD kann bei der Entwicklung eines XML-basierten Systems selbst festgelegt werden: XML trennt ebenso den Inhalt von der Darstellung, d.h. die Informationen über die Formatierung der Anzeige sind in der XML-Datei selbst nicht enthalten. Dies macht sich das beschriebene System ebenfalls zu nutze. Zudem steht mit XSLT zur Verarbeitung von XML-Dokumenten eine sehr vielseitige Formatierungs- bzw. Transformations-Sprache zur Verfügung.

In dieser Arbeit wird XML dazu verwendet, die Daten des Vorlesungsverzeichnisses zu repräsentieren. Erstellt wird die XML-Datei mit Hilfe von Abfragen und Makros in Microsoft Access. Die XML-Datei stellt die Basis für das Auskunftssystem dar.



## 5.2 Datentransformation und –selektion

Zur Transformation und Selektion der „rohen“ XML-Daten in ein vom Browser darstellbares und vom Benutzer „lesbares“ Format stehen zwei Möglichkeiten zur Verfügung: die Formatierung durch CSS oder die Transformation und Formatierung durch XSLT.<sup>25</sup>

### 5.2.1 Cascading Style Sheets (CSS)

Bei CSS handelt es sich um eine Formatierungssprache, die häufig in Verbindung mit HTML eingesetzt wird. Mit CSS können aber nicht nur HTML-Dateien, sondern auch XML-Dokumente für eine Ausgabe über den Browser formatiert werden.

Der Vorteil der CSS liegt in deren Einfachheit. Sie bieten die Möglichkeit bestimmten Bestandteilen eines Dokuments festgelegte Formatierungsmuster zuzuordnen. Dies gilt für den Einsatz bei HTML-Dokumenten ebenso wie bei der Verwendung mit XML-Dokumenten. Im Zusammenhang mit HTML wäre es vorstellbar, einem bestimmten Bereich (z.B. einer Auflistung mit `<ul>...</ul>` oder einer Tabelle) durch CSS festgelegte Formatierungsmuster zuzuweisen. Ähnlich ist die Verwendung von CSS mit XML zu verstehen. Man würde hier bestimmten Tags der XML-Datei eine festgelegte Darstellungsweise zuordnen. Für eine einfache formatierte Ausgabe stellen CSS also eine durchaus geeignete und einfache Lösung dar.

Komplexere Ausgaben lassen sich mit CSS in der aktuellen Version CSS Level 2 nicht erzeugen, was derzeit deren größten Nachteil darstellt. Dabei ist zum Beispiel an eine Sortierung der Ausgaben nach einem oder mehreren Werten oder an die Selektion bestimmter Werte zu denken.

Die Selektion von Werten und die Sortierung der Ausgabe stellen aber wichtige und grundlegende Anforderungen an das Auskunftssystem dar. Die Möglichkeiten, die CSS zur Formatierung von Dokumenten bieten, reichen in diesem Fall nicht aus.

---

<sup>25</sup> Nach Fung [2001, S. 6f] bestehe auch die Möglichkeit, XML-Dokumente mit Hilfe von Programmiersprachen wie C, C++ oder Java zu transformieren. Diese würden aber einen weit höheren Aufwand gegenüber Formatierungssprachen wie CSS und XSLT verursachen und daneben auch größere Programmdateien benötigen. Aus diesen Gründen wird hier auf diese Möglichkeiten nicht weiter eingegangen.

### 5.2.2 XSLT und XPath

Eine weitere Formatierungssprache für XML-Dokumente ist XSLT. XSLT wurde im November 1999 als W3C Recommendation veröffentlicht. XSLT ist komplexer als CSS, bietet dadurch aber auch mehr Möglichkeiten im Umgang mit XML-Dokumenten.

In einem XSLT-Stylesheet werden die Regeln festgelegt nach denen ein XML-Dokument transformiert wird. Ein solches XSLT-Stylesheet besteht aus mindestens einem Template. Ein XSLT-Prozessor vergleicht die Templates der verwendeten XSLT-Datei mit den Elementen des zu transformierenden XML-Dokuments. Mit Hilfe des XSLT-Prozessors wird durch die Templates eine Ausgabedatei erzeugt, für die unterschiedliche Dateiformate in Betracht kommen<sup>26</sup>.

Die Stärken von XSLT sind insbesondere

- die Fähigkeit ganz bestimmte Informationen aus einem XML-Dokument herauszusuchen zu können;
- die Möglichkeit bestimmte Regeln zum Ablauf des Transformationsprozesses festlegen zu können (zum Beispiel mit `xsl:if`, `xsl:choose`, `xsl:for-each`, etc.);
- die Fähigkeit sortierte Ausgaben erzeugen zu können;
- die Möglichkeit, durch die Übergabe von Parametern individuelle Transformationen vornehmen zu können;
- die Unterstützung zahlreicher Dateiformate und Ausgabemedien.

XSLT-Dokumente sind selbst XML-Dokumente, d.h. Dokumente in XML-Syntax oder mit anderen Worten: XSLT ist eine XML-Anwendung. Im Gegensatz zu CSS ist der Einsatz von XSLT damit auf XML-Dokumente beschränkt.

Im Zusammenhang mit XSLT kommt auch die XML Path Language (XPath) zum Einsatz. XPath wird im Gegensatz zu XSLT nicht in XML-Syntax formuliert. Sie dient dazu, Bereiche aus XML-Dokumenten zu erkennen.

---

<sup>26</sup> Eine vereinfachte Darstellung der Arbeitsweise von XSLT-Prozessoren wird in Abschnitt 5.2.3 (XSLT-Prozessoren) erläutert und durch Abbildung 5 veranschaulicht.

Mit Hilfe von XPath werden Knoten (engl. nodes) im zu transformierenden XML-Dokument ausgewählt. Innerhalb eines Dokuments unterscheidet XPath verschiedene Knoten: den root node, sowie element nodes, attribute nodes, text nodes, comment nodes, processing instruction nodes und namespace nodes.<sup>27</sup> Diese bezeichnen die verschiedenen Bestandteile einer XML-Datei. Die Bedingungen nach denen die Auswahl stattfindet, werden in der XSLT-Datei festgelegt.

XSLT verfügt damit über einen weitreichenden Funktionsumfang, der zur Realisierung des Prototyps ausreicht. Des weiteren sind XSLT und XPath – beides XML-Technologien – zum Einsatz mit XML geschaffen worden und daher besonders gut zu dessen Transformation geeignet. XSLT wird aus diesem Grund im beschriebenen Prototyp für die Transformation der XML-Dateien eingesetzt.

### 5.2.3 XSLT-Prozessoren

XSLT-Prozessoren sind Programme die ein eingelesenes XML-Quelldokument anhand eines XSLT-Stylesheets in ein Ausgabedokument transformieren. Das Quelldokument wird dabei nicht verändert, es wird lediglich dessen Struktur umgeformt und in eine durch das Stylesheet festgelegte Darstellungsform überführt.

Man unterscheidet verschiedene Arten von XSLT-Prozessoren:

1. In einen Web-Browser integrierte XSLT-Prozessoren
2. In einen Web-Server integrierte XSLT-Prozessoren
3. Command-Line Prozessoren (Kommandozeilenprozessoren)

Abbildung 5 veranschaulicht die Arbeitsweise eines XSLT-Prozessors. Der XSLT-Prozessor liest das XML-Quelldokument und das XSLT-Stylesheet ein und vergleicht die Templates der Stylesheet-Datei mit den Knoten des Quellbaums. Wird zu einem Knoten ein passendes Template gefunden wird dieses nach den dort definierten Angaben abgearbeitet und in den Zielknotenbaum überführt. Dieser wird nach der Verarbeitung in eine XML-Datei, HTML-Datei, Textdatei oder ein anders Format konvertiert. Werden dem XSLT-Prozessor beim Aufruf zusätzlich individuelle Parameter (z.B. Bezeichnung des Stu-

---

<sup>27</sup> Zur Bedeutung von Knoten im Zusammenhang mit XML und deren Dokumentstrukturen vgl.

diengangs oder das Fachsemester) mitgeteilt, können diese in die Abarbeitung des Stylesheets mit übernommen und berücksichtigt werden.

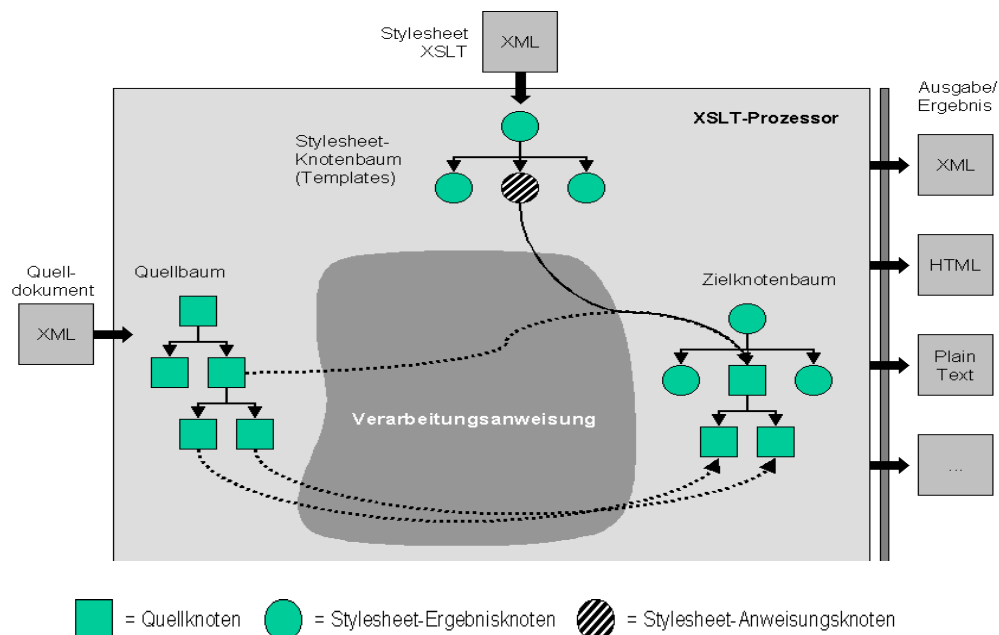


Abbildung 5: Vereinfachte Darstellung der Arbeitsweise eines XSLT-Prozessors<sup>28</sup>

In Web-Browser integrierte XSLT-Prozessoren werden auf dem Client, also auf dem Rechner des Benutzers ausgeführt. Dazu wird das Quelldokument auf den Rechner des Benutzers heruntergeladen und dort verarbeitet. Das Quelldokument muss in diesem Fall eine Referenz auf die XSLT-Datei enthalten (z.B.: `<?xml-stylesheet type="text/xsl" href="http://[Domain]/beispiel.xsl"?>`) damit der XSLT-Prozessor weiß, woher er die XSLT-Datei, mit der das Quelldokument zu verarbeiten ist, beziehen kann. Ein Beispiel für einen solchen XSLT-Prozessor ist der Microsoft XML-Parser (MSXML), der derzeit in Version 3.0 vorliegt.<sup>29</sup>

Bach [2000, S. 69ff]

<sup>28</sup> in Anlehnung an: Anderson et. al. [2000, S. 377]

<sup>29</sup> Version 4.0 ist als Technology Preview verfügbar.

Nachteil dieser Art von XSLT-Prozessoren ist insbesondere, dass alle Benutzer diesen auf ihrem System installiert haben müssen. Voraussetzung zur Nutzung ist auch, dass, wie im Fall von MSXML, ein aktueller Microsoft-Browser vorhanden sein muss (Internet Explorer Version 5.0 oder höher).

Wie bereits ausgeführt, lässt die aktuelle Situation im Fachbereich eine solche Lösung im Augenblick (noch) nicht zu. Ebenso müssten Benutzer, die von anderen Rechnern, z.B. von zu Hause auf das Auskunftssystem zugreifen den Internet Explorer in der Version 5.0 oder höher mit MSXML installiert haben. Dies ist aber nicht als voraussetzbar anzunehmen. Die Möglichkeit für einen Einsatz von MSXML steht also denkbar schlecht.

Die zweite Art von XSLT-Prozessoren, in Web-Server integrierte XSLT-Prozessoren, werden auf einem Server ausgeführt. Der Browser des Benutzers erhält vom Server bereits die Ausgabedatei z.B. in Form von HTML. Die Abhängigkeit vom Browsertyp entfällt dadurch.

Ein Beispiel für ein in einen Web-Server integrierten XSLT-Prozessor ist das Web Publishing Framework Cocoon aus dem Apache XML-Projekt. Die grundlegende Funktionsweise von Cocoon wurde bereits in Abschnitt 3.2.2 dargelegt. In Cocoon muss ein Command-Line Prozessor wie z.B. XT oder Apache Xalan und ein Parser integriert werden. Wie bereits erwähnt wurde, ist die Installation von Cocoon äußerst aufwendig und hat weitere Anpassungsarbeiten auf dem Server zur Folge.

Die dritte Art von XSLT-Prozessoren sind die Command-Line Prozessoren. Diese sind eigenständige Programme, sind also nicht in einen Client- oder Serverprozess integriert. Bei der Verarbeitung eines XML-Dokuments mit einem XSLT-Stylesheet wird der Prozessor in der Regel über die Kommandozeile aufgerufen, wobei diesem dabei üblicherweise der Pfad und der Name des Quelldokuments, des Ausgabedokuments und der Stylesheet-Datei mitgeteilt wird. Ebenso ist es bei den meisten dieser Prozessoren möglich, bei der Verarbeitung der XML-Datei mit dem Stylesheet individuelle Parameter zu übergeben.

Typische Command-Line Prozessoren sind Xalan<sup>30</sup> aus dem Apache XML-Projekt, XT<sup>31</sup> von James Clark, Sablotron<sup>32</sup> von Ginger Alliance Ltd. und SAXON<sup>33</sup> von Michael Kay.

---

<sup>30</sup> <http://xml.apache.org/xalan-c/>

<sup>31</sup> <http://www.jclark.com/xml/xt.html>

Bei der Verwendung eines Command-Line Prozessors innerhalb einer Web-Anwendung tritt das Problem auf, dass der Benutzer den Prozessor nicht über die Kommandozeile starten kann. Um diese Aufgabe zu übernehmen empfiehlt sich PHP oder ein CGI-Skript<sup>34</sup>. Über diese ist es möglich einen XSLT-Prozessor über die Kommandozeile aufzurufen. Für PHP spricht zudem, dass inzwischen eine XSLT-Erweiterung in PHP vorhanden ist, die den Sablotron-Prozessor in PHP integriert.

Laut PHP Handbuch [Bakken/Schmidt 2001; Abschnitt LXXXIX] ist die XSLT-Erweiterung derzeit noch in einem experimentellen Stadium, d.h. es ist unsicher ob alle Funktionen so beibehalten werden. Dennoch stellt die Erweiterung mit der Einbindung von Sablotron eine interessante Möglichkeit dar, zumal PHP an der Hochschule bereits in großem Umfang eingesetzt wird.

Bei ersten Tests mit Sablotron und Xalan musste festgestellt werden, dass Xalan Schwierigkeiten mit der Übernahme von Parametern mit deutschen Umlauten hat. Dieses Problem konnte auch nicht durch die Codierung der XML- und XSLT-Dateien mit dem Zeichensatz für westeuropäische Sprachen (ISO-8859-1) oder als 16-bit-Unicode (UTF-16) behoben werden. Die Verwendung von Sablotron ist in dieser Hinsicht weniger problematisch. Sablotron verarbeitet die deutschen Umlaute richtig.

Die beste der zur Verfügung stehenden Alternativen scheint die Kombination aus PHP mit dem Sablotron Prozessor sein. Ein clientseitig arbeitender Prozessor kommt nicht in Frage, da dies voraussetzt, dass der Browser des Nutzers XML-fähig ist. Bei serverseitiger Verarbeitung ist ein Webbrowser ausreichend, da nur HTML-Code dargestellt werden muss. Auch der Einsatz eines Web Publishing Framework wie Apache Cocoon ist im Grunde nicht vorstellbar, da, solange einfachere Lösungsmöglichkeiten bestehen, der Installations- und Konfigurationsaufwand bei der Realisierung des Prototyps nicht in einem angemessenen Verhältnis zum Nutzen steht. Der Einsatz von PHP wiederum ermöglicht, schon bekannte und im Fachbereich bewährte Technologien zu nutzen. Da PHP im Zusammen-

---

<sup>32</sup> [http://www.gingerall.com/charlie/ga/xml/p\\_sab.xml](http://www.gingerall.com/charlie/ga/xml/p_sab.xml)

<sup>33</sup> <http://saxon.sourceforge.net/>

<sup>34</sup> CGI (Common Gateway Interface) ist eine standardisierte Schnittstelle zum Datenaustausch zwischen Programmen auf dem Server und Browser. CGI-Skripte sind in der Regel in der Programmiersprache Perl geschrieben.

spiel mit Sablotron arbeitet, ist dessen Einsatz bei der Verwendung der XSLT-Erweiterung von PHP unumgänglich.

## **5.3 Datenpräsentation im Web**

Bisher wurden die Technologien beschrieben, die für die Erstellung der XML-Repräsentation der Stundenplandaten, die Erstellung der Stylesheet sowie zur Durchführung des Transformationsprozesses verwendet wurden. Nun fehlt noch ein letzter Schritt: die Darstellung des Ergebnisses im Web und die Führung des Nutzers durch das System.

### **5.3.1 PHP und Hypertext Markup Language (HTML)**

HTML bildet weiterhin den Standard zur Darstellung von Webseiten. Auch die in dieser Arbeit beschriebene Lösung setzt HTML ein. Zum einen wird es benützt um die nutzerspezifischen Parameterwerte zu erfassen, zum anderen wird HTML als Ausgabeformat für die transformierte XML-Datei verwendet.

Im ersten Fall wird HTML zusammen mit PHP angewendet. Mit Hilfe von PHP werden die verschiedenen Parameterwerte zum Beispiel auf Vollständigkeit überprüft. Dies ist notwendig um eine gültige Ausgabe zu erhalten, damit z.B. nicht ein Stundenplan ohne Angabe des Studiengangs generiert wird.

Der Transformationsprozess der XML-Datei verwendet HTML als Ausgabeformat um einem möglichst großen Nutzerkreis dienen zu können. HTML kann von jedem Web-Browser interpretiert werden. Dies ist, wie bereits erwähnt, auch der Grund, warum der Transformationsprozess bereits auf dem Server stattfindet.

### **5.3.2 Cascading Style Sheets (CSS)**

Cascading Style Sheets wurden bereits im Abschnitt 5.2.1 beschrieben als die Möglichkeiten zur Formatierung von XML-Dateien analysiert wurden. Dabei wurde festgestellt, dass diese für eine Darstellung von XML-Dokumenten im gegebenen Umfeld nicht ausreichten. Zur einheitlichen Formatierung von HTML-Dokumenten stellt CSS jedoch eine sehr verbreitete Technologie dar.

---

Im Bereich der Datenpräsentation im Web wird CSS dazu benutzt, ein einheitliches Layout in der Darstellung zu unterstützen. So wird mit CSS beispielsweise die Schriftgröße und Schriftart der Überschriften oder unterschiedlicher Termini einheitlich festgelegt.



## **6 Beschreibung der durchgeführten Arbeiten**

Grundlage für die Erstellung des Stundenplansystems für den Fachbereich IuK stellt, wie bereits erwähnt, die Analyse der Studien- und Prüfungsordnung (SPO) dar. Auf die daraus gewonnenen Erkenntnisse bauen die weiteren Schritte unmittelbar auf. Zuerst wurde aus der SPO des Fachbereichs die strukturellen und organisatorischen Unterschiede der Studiengänge gewonnen, die im weiteren Verlauf der Arbeiten berücksichtigt werden mussten.

### **6.1 Analyse der Studien- und Prüfungsordnung**

Der Fachbereich IuK der HdM bietet verschiedene Studiengänge an. Die Diplomstudiengänge Informationsmanagement (IM), Öffentliche Bibliotheken (ÖB) und Wissenschaftliche Bibliotheken (WB) sind bis 2003 auslaufende Studiengänge. Der Studiengang IM wurde zum Wintersemester 2000/01 durch den Diplomstudiengang Informationswirtschaft (IW) ersetzt. Die Studiengänge ÖB und WB wurden zum selben Termin durch zwei Studiengänge im Bereich Bibliotheks- und Medienmanagement (BM) abgelöst. Diese werden mit unterschiedlichen Schwerpunkten durchgeführt. Der Diplomstudiengang Bibliotheks- und Medienmanagement (BMD) hat die beiden Bereiche Öffentliche Bibliotheken und Wissenschaftliche Bibliotheken zum Schwerpunkt, der gleichnamige Bachelorstudiengang (BMB) hat den Bereich Medien als Schwerpunktthema. Zum Wintersemester 2001/02 beginnt außerdem der neue Bachelorstudiengang Informationsdesign (ID).

Zusätzlich zu diesen grundständigen Studiengängen werden zwei Aufbaustudiengänge durchgeführt. Der Masterstudiengang Bibliotheks- und Medienmanagement (Master BM) wird zur Zeit zum ersten Mal durchgeführt. Bereits zum Sommersemester 2000 startete der Masterstudiengang Informationswirtschaft (Master IW).

Einen Sonderfall stellt das Musikbibliothekarische Zusatzstudium dar, das der Fachbereich ebenfalls anbietet. Dieses wird als Fern- und Blockstudium durchgeführt und ist nicht in der Stundenplandatenbank erfasst. Aus diesem Grund wird es im weiteren Verlauf außer Acht gelassen.

Die Struktur der Studiengänge wird in der SPO festgelegt. Die SPO des Jahres 1997 regelt die Studiengänge IM, ÖB und WB. Die anderen Studiengänge werden durch die Studien- und Prüfungsordnungen des Jahres 2001 festgelegt.

Für eine individuelle Ausgabe von Stundenplänen für Studenten spielt in hohem Masse die Kombination der gewählten Wahlpflichtfächer eine Rolle. Dabei ist wichtig, ab welchem Semester eine Wahlmöglichkeit besteht, wie viele Wahlpflichtfächer angeboten werden, und wie viele aus diesem Angebot gewählt werden müssen. Die folgenden Abschnitte analysieren die jeweiligen Studien- und Prüfungsordnungen in Hinblick auf diese Anforderungen.

### **6.1.1 Studiengänge IM, ÖB und WB**

Laut der SPO des Jahres 1997 [HBI 1997] wurden die Studiengänge IM, ÖB und WB im Grundstudium, welches das 1. und 2. Fachsemester umfasst, gemeinsam unterrichtet. Da es sich um auslaufende Studiengänge handelt ist dieser Fall nicht mehr zu berücksichtigen. Alle Studierenden dieser Studiengänge sind in der Durchführung bereits über das Grundstudium hinaus.

Ab dem 3. Fachsemester finden die Veranstaltungen laut SPO nach Studiengängen getrennt statt. Zusätzlich kämen ab dem 2. Studiensemester Wahlpflichtfächer hinzu, von denen die Studenten aus drei unterschiedlichen Bereichen je ein Wahlpflichtfach wählen müssten. Dabei gelte zu berücksichtigen, dass einige dieser Wahlpflichtfächer nur für einen Studiengang angeboten würden, andere jedoch für zwei oder alle drei Studiengänge.

Aus dem ersten Wahlpflichtbereich (Medien und Information) seien im Laufe des Studiums 16 SWS zu belegen, aus dem zweiten Bereich (Spezielle Arbeitsfelder/Langzeitprogramme) zwölf SWS und aus dem dritten Bereich (Spezielle Arbeitsfelder/Kurzzeitprogramme) acht SWS. Diese Regelung gilt für alle drei Studiengänge gleichermaßen. Im Umfeld der Wahlpflichtprogramme kann es zudem vorkommen, dass dort angebotene Fächer, als *pflicht* angeboten werden und daher zu belegen sind, andere als *wahlfrei* gekennzeichnet sind und nicht verpflichtend zu belegen sind.

### 6.1.2 Studiengang IW

Nach §35 der SPO für die Diplom- und Bachelorstudiengänge [HdM 2001a] gelten für Studierende im Studiengang IW folgende Regeln:

*„(3) Das Hauptstudium umfasst die [auf das 2. Semester] folgenden fünf Studiensemester [...]. Es besteht aus einem Pflichtbereich im Umfang von 64 SWS und einem Wahlpflichtbereich im Umfang von 30 SWS.*

*(4) Im Wahlpflichtbereich sind zwei Module im Umfang von jeweils 15 SWS zu belegen [...].“*

Im Gegensatz zu den zuvor beschriebenen Studiengängen IM, ÖB und WB beginnen die Veranstaltungen im Wahlpflichtbereich erst im Hauptstudium. Außerdem werden die beiden zu wählenden Module nicht aus unterschiedlichen Angeboten, sondern aus einem Angebot gewählt. Nach der SPO [HdM 2001a, §35 Tabelle 4] sind aus den drei angebotenen Wahlpflichtmodulen Management, Informationstechnik und Informationswirtschaft zwei Module auszuwählen.

### 6.1.3 Studiengänge BMB und BMD

Ähnliches trifft auch auf die Studenten der Studiengänge BMB und BMD zu, deren Wahlfreiheit jedoch noch größer ist. Laut einem internen Papier der Hochschule [Hütter 2001] werden die sechs Wahlpflichtbereiche für die Bachelor- und Diplomstudiengänge Bibliotheks- und Medienmanagement gemeinsam angeboten.

Die SPO regelt wie die einzelnen Studiengänge die Veranstaltungen zu belegen haben. Für Studenten des Diplomstudiengangs gilt die in §33 festgelegte Regelung:

*„(3) Das Hauptstudium umfasst die [auf das 2. Semester] folgenden fünf Studiensemester [...]. Es besteht aus einem Pflichtbereich im Umfang von 53 SWS und einem Wahlpflichtbereich im Umfang von 44 SWS.*

*(4) [...] Aus dem Wahlpflichtbereich 1 und/oder dem Wahlpflichtbereich 2 sind Module im Umfang von insgesamt mindestens 20 SWS zu belegen. [...]“ [HdM 2001a, §33]*

Diese Bestimmungen besagen, dass die Studenten aus den Wahlpflichtbereichen 1 und 2 insgesamt 20 SWS belegen müssen. Dabei können sie auch alle 20 Semesterwochenstunden

in einem der beiden Bereiche ableisten. Die übrigen 24 SWS können dann aus allen sechs Bereichen gewählt werden.

Für den Studiengang BMB gelten annähernd die gleichen Regeln. Unterschiede finden sich jedoch in der Stundenzahl und der Wahl der Wahlpflichtmodule:

*„(3) Das Hauptstudium umfasst die [auf das 2. Semester] folgenden fünf Studiensemester [...]. Es besteht aus einem Pflichtbereich im Umfang von 51 SWS und einem Wahlpflichtbereich im Umfang von 44 SWS.*

*(4) [...] Aus dem Wahlpflichtbereich 3 sind Module im Umfang von mindestens 24 SWS zu belegen. [...]“ [HdM 2001a, §34]*

BMB-Studenten müssen demnach ebenso 44 SWS aus dem Wahlpflichtbereichen belegen. 24 SWS davon im Bereich 3, den Rest können sie aus allen sechs Bereichen frei auswählen.

#### **6.1.4 Studiengang ID**

Der Studiengang ID ist ähnlich strukturiert wie der Studiengang IW. Über die Wahlmöglichkeiten der ID-Studierenden sagt die SPO [HdM 2001a, §36] folgendes aus:

*„(3) Das Hauptstudium umfasst die [auf das 2. Semester] folgenden fünf Studiensemester [...]. Es besteht aus einem Pflichtbereich im Umfang von 70 SWS und einem Wahlpflichtbereich im Umfang von 20 SWS.*

*(4) Im Wahlpflichtbereich sind zwei Module im Umfang von jeweils 10 SWS zu belegen [...]“*

Laut HdM [2001a, §36 Tabelle 4] haben die Studenten die Wahl aus vier Wahlmodulen. Neben den beiden gewählten Modulen sind keine weiteren Wahlpflichtfächer zu belegen.

Der Studiengang ID startet zum Wintersemester 2001/02 zum ersten Mal. Eine Entscheidung für die Wahlmodule wirkt sich erst zum Wintersemester 2002/03 aus. Daher ist zu überlegen, ob die Wahlmodule im hier beschriebenen System zum aktuellen Zeitpunkt überhaupt vorgesehen werden müssen.

#### **6.1.5 Masterstudiengänge BM und IW**

Die Masterstudiengänge BM und IW unterscheiden sich in hohem Maße von den zuvor beschriebenen Studiengängen. Laut SPO für die Masterstudiengänge [HdM 2001b] beträgt

die Regelstudienzeit für den Studiengang Master IW drei Semester, die für den als Teilzeitstudium angebotenen Master BM fünf Semester. Demnach sei jeweils das letzte, also das dritte bzw. fünfte Studiensemester als praktische Studienzeit vorgesehen. Vorlesungen finden demzufolge für Studenten des Studiengangs Master IW nur zwei, für Studenten des Studiengangs Master BM nur vier Semester statt.

Die SPO [HdM 2001b] schreibt für beide Studiengänge vier Module als Pflicht vor. Insgesamt werden jeweils acht Module angeboten. Die Dauer der Module erstreckt sich über die gesamte theoretische Studienzeit. In diesem Sinne gibt es also streng genommen für die Masterstudiengänge keine Pflichtfächer, sondern lediglich Wahlpflichtfächer.

### 6.1.6 Zusammenfassung

Die Analyse der Studien- und Prüfungsordnungen ergibt eine Vielzahl unterschiedlicher Wahlmöglichkeiten und Vorschriften. Tabelle 1 fasst diese kompakt zusammen.

Tabelle 1: Überblick über die Wahlmöglichkeiten in den Studiengängen

Studien- gang	Wahlmo- dule ab	Anzahl Module	Bemerkungen, Besonderheiten
IM, ÖB, WB	frühestens 2. Semester	3	Je ein Fach aus 3 unterschiedlichen Angebotsgruppen mit 16, 12 und 8 SWS; innerhalb der Wahlpflichtfächer teils wahlfreie Veranstaltungen, teils Pflichtveranstaltungen
IW	3. Semester	2	Von 3 Modulen sind 2 komplett zu belegen
ID	3. Semester	2	Von 4 Modulen sind 2 komplett zu belegen
BMB	3. Semester	bis zu 6	Aus Modul 3 (Medien) sind 24 SWS zu belegen, weitere 20 SWS sind aus allen 6 Modulen frei wählbar.
BMD	3. Semester	bis zu 6	Aus Modul 1 (Öffentliche Bibliotheken) und/oder Modul 2 (Wissenschaftliche Bibliotheken) sind 20 SWS zu belegen, weitere 24 SWS sind aus allen 6

Studien- gang	Wahlmo- dule ab	Anzahl Module	Bemerkungen, Besonderheiten
			Modulen frei wählbar.
Master BM, Master IW	1. Semester	4	Von 8 Modulen sind 4 komplett zu belegen

## 6.2 Analyse der Stundenplandaten

Nach der Analyse der Studien- und Prüfungsordnungen mussten die Inhalte der Datenbank analysiert werden. Da die Erstellung der XML-Datei später mit Hilfe von MS Access durchgeführt wird, ist ein Import der Datenbank bereits jetzt sinnvoll.

### 6.2.1 Konfiguration und Import der Datenquelle

Der Import der Datenbank erfolgt mit Hilfe von MyODBC. Der MyODBC-Treiber ist kostenlos aus dem Internet herunterladbar<sup>35</sup>.

In Abbildung 6 wird die Konfiguration des MyODBC-Treibers dargestellt. Über die Systemsteuerung werden die *ODBC-Datenquellen* aufgerufen. Es erscheint das in Abbildung 6 links dargestellte Fenster *ODBC-Datenquellen-Administrator*. Um die Stundenplandatenbank in Access importieren zu können, muss nun die Datenquelle konfiguriert werden. Hierzu wählt man unter Benutzer-DSN „Hinzufügen...“ und wählt als Treiber *MySQL* aus. Über die Schaltfläche „Fertig stellen“ erhält man die Treiberkonfiguration (in der Abbildung rechts dargestellt). Dort sind folgende Einstellungen zu treffen:

- Windows DSN name: Unter diesem Namen kann die Datenbank später angesprochen werden. Der Name ist frei wählbar.
- MySQL host (name or IP): Den Namen oder die IP-Adresse des Hosts, auf dem sich die Datenbank befindet. In diesem Fall: *v.bdm-stuttgart.de*
- MySQL database name: Den Namen der Datenbank. In diesem Fall: *ms2001*

---

<sup>35</sup> <http://www.mysql.org>

- User, Password: Den MySQL-Benutzernamen und das zugehörige Passwort. Hier für den Benutzernamen *klispa*

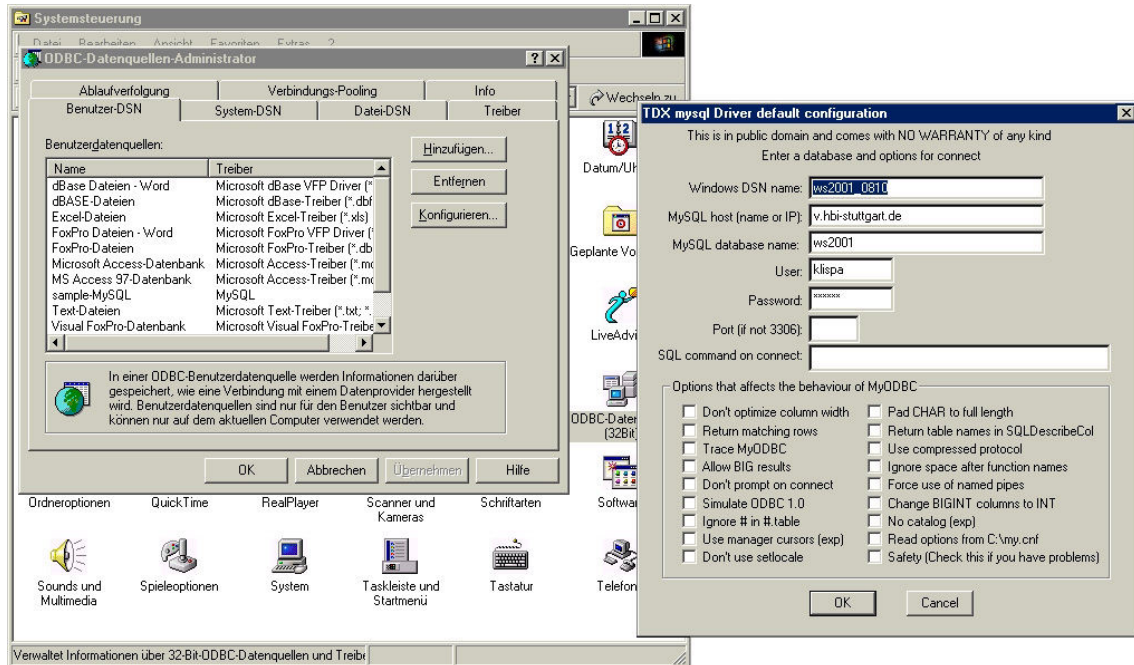


Abbildung 6: Konfiguration des MyODBC-Treibers

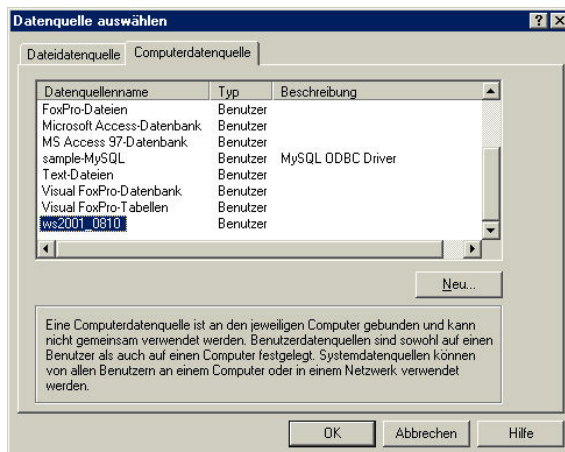


Abbildung 7: Auswahl der Datenquelle

Im Anschluss an die Konfiguration der Datenquelle kann die Datenbank in MS Access importiert werden. Dieser Vorgang wird in MS Access selbst durchgeführt. Über „Datei / Externe Daten / Importieren...“ wird das Fenster zum Importieren von Datenbanken geöffnet. Dort kann über die Auswahl *ODBC-Datenbanken* im Eingabefeld *Datentyp* die zuvor konfi-

gurierte Datenquelle aufgerufen werden. Unter *Computerdatenquelle* befindet sich die Datenquelle unter dem bereits festgelegten Windows DSN Name (Abbildung 7 zeigt diese Auswahl).

Der Import geschieht über die Schaltfläche „OK“ und die nachfolgende Auswahl der zu importierenden Tabellen. Da die Daten zunächst analysiert werden, müssen alle Tabellen importiert werden. Über die Schaltfläche „OK“ werden die ausgewählten Tabellen in die geöffnete Datenbank importiert.

### 6.2.2 Analyse des Datenbestandes

Um aus der Datenbank eine XML-Repräsentation herstellen zu können, muss der vorliegende Datenbestand zunächst analysiert werden. Dabei interessiert vor allem, welche Bedeutung die einzelnen Tabellen und deren Felder haben sowie in welcher Weise die Tabellen und deren Inhalt zueinander in Verbindung stehen.

#### Die Tabelle *erfass*

Die Tabelle *erfass* enthält alle Fächer, die im betreffenden Semester durchgeführt werden. Der Primärschlüssel dieser Tabelle ist das Feld *sigel*. Über dieses wird jedes Fach<sup>36</sup> eindeutig identifiziert. Das Feld *titel* enthält den Namen des Fachs, wie es später im Vorlesungsverzeichnis erscheint. Das Feld *art* gibt an, ob es sich um ein Pflicht-, Wahl(pflicht)fach oder um eine fakultative Lehrveranstaltung handelt. Bei den Einträgen der Masterstudiengänge lautet der Eintrag immer „Pflichtfach“, obwohl es streng genommen Fächer in Wahlbereichen sind. Fakultative Lehrveranstaltungen werden für alle Studiengänge angeboten und können zusätzlich zu den Veranstaltungen in Pflicht- und Wahlpflichtfächern besucht werden.

Bei den Studiengängen IM, ÖB und WB wird, wie bereits oben erwähnt, innerhalb des Wahlpflichtbereichs zusätzlich zwischen verpflichtenden und wahlfreien Veranstaltungen unterschieden. Ebenso wird bei diesen Studiengängen jedes Wahlpflichtfach aus einem

---

<sup>36</sup> Unter „Fach“ ist ein, innerhalb eines Moduls stattfindendes Fach zu verstehen. Ein Fach kann aus einer oder mehreren Gruppen, ergo Terminen, bestehen, ein Modul ist in der Regel aus mehreren Fächern aufgebaut (mindestens jedoch aus einem Fach). In der beschriebenen Datenbank ist „Fach“ jedoch bedeutungsgleich mit „Modul“! In den Tabellen *erfass* und *anrechenbarkeit* taucht das Feld *Fach* als Fremdschlüssel der Tabelle *modul* auf.



eigenen Bereich gewählt. Um dies in der Datenbank darzustellen, sind die Felder `WArtAlt` und `WArt7Alt` vorhanden. `WArtAlt` gibt Auskunft ob das Fach wahlfrei ist, `WArt7Alt` gibt den Bereich an, aus dem das Wahlpflichtfach gewählt wurde. Das Feld `Fach` enthält eine Nummer, die als Fremdschlüssel auf den Primärschlüssel der Tabelle `Modul` verweist und somit das Modul angibt, welchem das Fach zugeordnet ist. Das Feld `Kommentar` enthält Hinweise für die Planung der Veranstaltung durch das Stundenplanamt. `Anmerkung` gibt Hinweise zur Durchführung der Veranstaltungen, wie sie auch im Vorlesungsverzeichnis dargestellt werden. `Semester` gibt an, für welches Studiensemester das Fach durchgeführt wird. Das Feld `Studiengang` enthält keine Daten. Die Angabe des Studiengangs befindet sich in den Tabellen `anrechenbarkeit`, `fachzuord`, `modul` und `studiengang`. Als Feldwerte werden die Studiengangskürzel, wie in Abschnitt 6.1 eingeführt, verwendet. Die Bezeichnung BM steht in der Datenbank für Lehrveranstaltungen, die sowohl für den Studiengang BMB als auch für den Studiengang BMD durchgeführt werden. Das Feld `raumart` bezeichnet den Raumtyp, also z.B. ob es sich um einen Hörsaal, einen PC-Pool oder um einen Seminarraum handelt.

## Die Tabelle `lv`

Die Tabelle `lv` enthält Informationen zur räumlichen und zeitlichen Durchführung der Veranstaltungen. Über das Feld `sigel` ist es mit der Tabelle `erfass` verknüpft. Anhand des Felds `Nummer` erfährt man, ob die Lehrveranstaltung in mehreren Gruppen durchgeführt wird. Ist ein Sigel nur einmal in dieser Tabelle aufgeführt, gibt es nur eine Gruppe (der Wert des Felds `Nummer` ist dann „0“). Bei mehreren Gruppen sind für ein Sigel mehrere Datensätze vorhanden. Im Feld `Nummer` wird dann die Anzahl der Gruppen durchgezählt (beginnend mit „0“).

Das Feld `Gruppe` gibt den Namen der Gruppe an, wie er im Vorlesungsverzeichnis veröffentlicht wird. Das Feld `Tag` enthält den Wochentag, an dem der Termin stattfindet, das Feld `zeit` beinhaltet die Anfangszeit der Lehrveranstaltung. Der Tag ist in der Form „mo“, „di“, „mi“, „do“, „fr“ und „sa“ angegeben. Die Anfangszeit ist als Zahl eingetragen, wobei der Feldwert „1“ für die Anfangszeit 8:15 Uhr, der Feldwert „2“ für 9:15 Uhr usw. steht. Für nicht feststehende Termine ist das Feld `Tag` leer. Das Feld `zeit` hat in diesem Fall den Wert „0“ oder ist ebenfalls leer.

Das Feld `Raum` enthält die Nummer des Raums, in dem die Lehrveranstaltung stattfindet. Durch das Feld `DozentG` wird die Lehrkraft angegeben. Falls eine Veranstaltung von mehreren Lehrkräften durchgeführt wird, werden diese alle in das Feld `DozentG` eingetragen.

### Die Tabelle *modul*

Die Tabelle `modul` enthält alle Module, in denen Veranstaltungen durchgeführt werden. Das Feld `ID` ist der Primärschlüssel dieser Tabelle. Über `ID` ist diese Tabelle mit dem Feld `fach` der Tabelle `erfass` verknüpft. Das Feld `Modul` enthält die Bezeichnung des Moduls, die Felder `Studiengang` und `Semester` beinhalten den Studiengang und das Fachsemester, für die das Modul durchgeführt wird. `Art` enthält Angaben über Wahlmodule.

### Die Tabelle *studiengang*

Die Tabelle `studiengang` enthält die beiden Felder `Sigel` und `Studiengang`. Der Primärschlüssel dieser Tabelle besteht aus deren Kombination. In dieser Tabelle wird angegeben, für welche Studiengänge das durch das `Sigel` identifizierte Fach durchgeführt wird.

### Die Tabelle *anrechenbarkeit*

Wie bereits anhand der Tabelle `studiengang` erklärt wurde, ist es möglich, dass Fächer für mehrere Studiengänge gemeinsam durchgeführt werden. Es ist außerdem denkbar, dass ein Fach für unterschiedliche Wahlmodule anrechenbar ist. Diese Tatsache wird durch die Tabelle `anrechenbarkeit` berücksichtigt. Die Tabelle ist über das Feld `sigel` mit der Tabelle `erfass` verknüpft. Das Feld `Fach` enthält das Modul, für das das Fach ebenfalls anrechenbar ist. `Studiengang` und `Semester` enthalten die zugehörigen Daten über Studiengang und Studiensemester. Weitere Felder geben Auskunft über Art der anrechenbaren Wahlfächer, die im Zusammenhang mit der weiteren Verwendung der Daten nicht von Bedeutung sind.

### Die Tabelle *fachzuord*

Die Tabelle `fachzuord` enthält die, für die im Stundenplanamt durchgeführte Erstellung der Stundenpläne notwendige Auflösung der Bezeichnung BM zu BMB und BMD. Darüber hinaus sind in dieser Tabelle Informationen zur Zuordnung von anrechenbaren Veranstaltungen enthalten. Für die Generierung von Terminplänen auf Basis von XML wird diese Tabelle nicht benötigt.

### Die Tabelle *dozenten*

Die Tabelle *dozenten* enthält weitere Informationen über die Durchführung der Lehrveranstaltungen. Über *sigel* ist die Verknüpfung mit der Tabelle *erfass* hergestellt. Der Primärschlüssel dieser Tabelle ist die Kombination aus den Feldern *sigel* und *dozent*. Das Feld *dozent* beinhaltet als Fremdschlüssel die Nummer des Dozenten und ist somit wiederum mit der Tabelle *dozadr* verknüpft. In *Durchführung* wird die Dauer der Veranstaltung in Stunden angegeben. Das Feld *stud\_sws* enthält die Anzahl der SWS, die auf diese Veranstaltung für die Studenten anrechenbar sind. *doz\_sws* enthält die Anzahl der Deputatsstunden, die der Dozent für die Durchführung der Veranstaltung angerechnet bekommt. Die weiteren Felder dieser Tabelle haben lediglich für die Planung der Termine im Stundenplanamt Bedeutung.

### Die Tabelle *dozadr*

Die Tabelle *dozadr* verzeichnet alle Lehrkräfte. Über das Feld *nummer* sind diese eindeutig identifiziert. Außerdem umfasst die Tabelle *dozadr* alle Kontaktdaten der Lehrkräfte wie z.B. Straße, Ort, Telefon, E-Mail, etc.

### Die Tabellen *it* und *ita*

Die Tabellen *it* und *ita* enthalten Informationen über die benötigte IT-Ausstattung in den Lehrveranstaltungen. Diese Informationen sind für die Ausgabe von Terminplänen nicht von Bedeutung. Sie betreffen vielmehr die Planung des Stundenplans und den IT-Service des Fachbereichs.

### Zusammenfassung

Die „zentrale“ Tabelle der Datenbank ist die Tabelle *erfass*. Sie enthält die durchzuführenden Fächer. Über den Fremdschlüssel *fach* ist das Modul bestimmt. Die Studiengänge, für die das Fach abgehalten wird, können aus der Tabelle *studiengang* bezogen werden. Die Tabellen *lv* und *dozenten* sind für die Durchführung der einzelnen Veranstaltungen innerhalb der Fächer wichtig. Sie enthalten Informationen über Zeitpunkt, Dauer, Raum, Lehrkraft und Gruppe. Die Tabelle *anrechenbarkeit* wird für die Zuordnung von Fächern auf mehrere Module benötigt.

Die fakultativen Lehrveranstaltungen sind getrennt von den einer Gruppe von Studenten fest zurechenbaren Lehrveranstaltungen der Pflicht- und Wahlpflichtfächer zu betrachten. Deren Eigenheit ist, dass sie für alle Studenten offen sind.

Die Beschreibung der Erstellung einer XML-Repräsentanz der für die Ausgabe von Terminplänen wichtigen Daten, wird nach der Festlegung der XML-Dokumentstruktur in Abschnitt 6.4 geschildert.

### 6.3 XML-Dokumentstruktur

In Abschnitt 4.2 wurden die inhaltlichen Anforderungen an die Ausgabe von benutzerindividuellen Terminplänen bestimmt. Dabei handelt es sich in erster Linie um die Auswahl der Daten, die dem Benutzer in der Ergebnisanzeige ausgegeben werden sollen und die zur Verarbeitung der individuellen Anfragen benötigt werden.

Auf Basis der inhaltlichen Anforderungen lässt sich eine XML-Dokumentstruktur, Document Type Definition (DTD), erstellen.

Eine für die XML-Darstellung sehr naheliegende Struktur ist die hierarchische Abbildung der Stundenplandaten. Darunter ist eine verschachtelte DTD zu verstehen, die den Stundenplan in hierarchische Einheiten zerlegt. Als Beispiel einer solchen Abbildung könnte folgende DTD dienen:

```
<!ELEMENT plan (studiengang)+>
  <!ELEMENT studiengang (stud_bezeichnung, semester+)>
    <!ELEMENT stud_bezeichnung (#PCDATA)>
    <!ELEMENT semster (sem_bezeichnung, modul+)>
      <!ELEMENT sem_bezeichnung (#PCDATA)>
      <!ELEMENT modul (modul_bezeichnung, fach+)>
        <!ELEMENT modul_bezeichnung (#PCDATA)>
        <!ELEMENT fach (sigel, titel, sws, anmerkung?, art, wahlartalt?,
durchfuehrung, termin+)>
          <!ELEMENT sigel (#PCDATA)>
          <!ELEMENT titel (#PCDATA)>
          <!ELEMENT sws (#PCDATA)>
          <!ELEMENT anmerkung (#PCDATA)>
          <!ELEMENT art (#PCDATA)>
          <!ELEMENT wahlartalt (#PCDATA)>
          <!ELEMENT durchfuehrung (#PCDATA)>
```

```
<!ELEMENT termin (gruppe, tag, zeit, dozent, raum?)>
  <!ELEMENT gruppe (#PCDATA)>
  <!ELEMENT tag (#PCDATA)>
  <!ELEMENT zeit (#PCDATA)>
  <!ELEMENT dozent (#PCDATA)>
  <!ELEMENT raum (#PCDATA)>
```

Diese DTD beruht auf den folgenden Überlegungen:

- In einem Vorlesungsverzeichnis gibt es mehrere Studiengänge (einen oder mehrere Studiengänge).
- Ein Studiengang hat eine Bezeichnung und ein oder mehrere Studiensemester.
- Ein Studiensemester hat eine Bezeichnung und Vorlesungen in mindestens einem Modul.
- Jedes Modul hat eine Bezeichnung. Ein Modul besteht aus mindestens einem Fach.
- Jedes Fach hat genau ein Sigel, einen Titel, eine Anzahl an SWS, möglicherweise eine Anmerkung. Es hat eine Art, eine Durchführungsdauer und wird je nach Gruppenanzahl an einem oder mehreren Terminen unterrichtet. Das Element `wahlartalt` berücksichtigt den Umstand, dass bei Wahlpflichtfächern der Studiengänge IM, ÖB und WB eine weitere Differenzierung zwischen Wahl und Pflicht notwendig ist (vgl. Seite 42).
- Jeder Termin wird für genau eine Gruppe an einem bestimmten Tag, zu genau einer Zeit von einem Dozenten durchgeführt und kann in einem Raum an der Hochschule durchgeführt werden (oder außerhalb bzw. als virtuelle Lehrveranstaltung).

Eine beispielhafte XML-Datei gemäß dieser DTD sähe folgendermaßen aus:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE plan SYSTEM "[URI der verwendeten DTD]">
<plan>
  <studiengang>
    <stud_bezeichnung>BMD</stud_bezeichnung>
    <semester>
      <sem_bezeichnung>1</sem_bezeichnung>
    <modul>
      <modul_bezeichnung>Informationsressourcen 1</modul_bezeichnung>
```

```
<fach>
  <sigel>5</sigel>
  <titel>Seminar</titel>
  <sws>2</sws>
  <anmerkung/>
  <art>Wahlpflichtbereich</art>

  <wahlartalt/>
  <durchfuehrung>2</durchfuehrung>
  <termin>
    <gruppe>BMD 1abc</gruppe>

    <tag>mi</tag>
    <zeit>5</zeit>
    <dozent>Biener</dozent>
    <raum>011</raum>
  </termin>
  <termin>
    <gruppe>BMD 2abc</gruppe>

    <tag>mo</tag>
    <zeit>5</zeit>
    <dozent>Hoffmann</dozent>
    <raum>011</raum>
  </termin>
</fach>
</modul>
</semester>
</studiengang>
</plan>
```

Bei der Festlegung der DTD ist bereits jetzt zu bedenken, dass die Erstellung dieser Datei mit Abfragen aus der Datenbank in MS Access durchgeführt wird. Bei der Formulierung dieser Abfragen ist man vom Datenbankdesign der Datenquelle abhängig und muss die dadurch gegebenenfalls vorhandenen Einschränkungen berücksichtigen.

Bei der Erstellung der DTD sollte außerdem beachtet werden, dass Lehrveranstaltungen nicht auf eine bestimmte Studentengruppe beschränkt sein müssen. Die Lehrveranstaltungen können, wie bereits dargestellt,

- für verschiedene Studiengänge gemeinsam abgehalten werden;
- für mehrere Studiensemester durchgeführt werden;

- für unterschiedliche Module anrechenbar sein.

Wie an dieser Stelle noch einmal betont werden soll, ist es sogar möglich, dass alle drei oben erwähnten Fälle bei *einer* Lehrveranstaltung auftreten können.

Eine weitere Möglichkeit die Daten darzustellen ist die nicht-hierarchische Abbildung. Bei dieser Herangehensweise wird jeder Termin als eigene Lehrveranstaltung angesehen und beinhaltet somit alle notwendigen Daten. Die folgende DTD zeigt die nicht-hierarchische Abbildung der Stundenplandaten:

```
<!ELEMENT plan (lv)+>
  <!ELEMENT lv (sigel, semester, studiengang, modul, titel, dozent,
durchfuehrung, sws, anmerkung?, tag, zeit, raum?, gruppe?, wahlartalt?,
art)>
    <!ELEMENT sigel (#PCDATA)>
    <!ELEMENT semester (#PCDATA)>
    <!ELEMENT studiengang (#PCDATA)>
    <!ELEMENT modul (#PCDATA)>
    <!ELEMENT titel (#PCDATA)>
    <!ELEMENT dozent (#PCDATA)>
    <!ELEMENT durchfuehrung (#PCDATA)>
    <!ELEMENT sws (#PCDATA)>
    <!ELEMENT anmerkung (#PCDATA)>
    <!ELEMENT tag (#PCDATA)>
    <!ELEMENT zeit (#PCDATA)>
    <!ELEMENT raum (#PCDATA)>
    <!ELEMENT gruppe (#PCDATA)>
    <!ELEMENT wahlartalt (#PCDATA)>
    <!ELEMENT art (#PCDATA)>
```

Bei einer Ausrichtung auf diese DTD hätte die oben beschriebene XML-Datei folgende Struktur:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE plan SYSTEM "[URI der verwendeten DTD]">
<plan>
  <lv>
    <sigel>5</sigel>
    <semester>1</semester>
    <studiengang>BMD</studiengang>
    <modul>Informationsressourcen 1</modul>
    <titel>Seminar</titel>
```

```

<dozent>Biener</dozent>
<durchfuehrung>2</durchfuehrung>
<sws>2</sws>
<anmerkung/>
<tag>mi</tag>
<zeit>5</zeit>

<raum>011</raum>
<gruppe>BMD 1abc</gruppe>
<wahlartalt/>
<art>Wahlpflichtbereich</art>

</lv>
<lv>
  <sigel>5</sigel>
  <semester>1</semester>
  <studiengang>BMD</studiengang>
  <modul>Informationsressourcen 1</modul>
  <titel>Seminar</titel>

  <dozent>Hoffmann</dozent>
  <durchfuehrung>2</durchfuehrung>
  <sws>2</sws>
  <anmerkung/>

  <tag>mo</tag>
  <zeit>5</zeit>
  <raum>011</raum>
  <gruppe>BMD 2abc</gruppe>
  <wahlartalt/>

  <art>Wahlpflichtbereich</art>
</lv>
</plan>

```

Die Dozenten- und Raumauskunft benötigt weniger Angaben als die in dieser DTD vorgesehenen. Bei unterschiedlichen XML-Repräsentationen für Stundenplanauskunft und Dozenten- und Raumauskunft wäre dann auch eine leicht verkürzte DTD ausreichend.

Gegenüber der hierarchischen Abbildung der Daten liegt der Nachteil der nicht-hierarchischen Abbildung in einer voraussichtlich etwas größeren XML-Datei. Den Umstand, dass mehrfach anrechenbare Veranstaltungen wiederholt aufgeführt werden müssen, kann keine der beiden Alternativen beheben.



Die vielfachen Möglichkeiten einer unterschiedlichen Anrechenbarkeit macht die Erstellung einer hierarchischen Abbildung der Daten in einer XML-Datei mit Hilfe von Abfragen in MS Access sehr schwierig. Die nicht-hierarchische Abbildung der Daten ist mit Access leichter zu realisieren. Wie bereits ausgeführt wurde, wird durch die Verarbeitung der XML-Datei auf dem Server ein Herunterladen des XML-Dokuments vermieden. Die etwas größere XML-Datei in der nicht-hierarchischen Abbildung fällt also kaum ins Gewicht und verlangsamt den Verarbeitungsprozess nur in einem sehr geringen Maße. Nach Abwägung der Vor- und Nachteile fiel die Entscheidung auf die nicht-hierarchische Lösung, deren Erstellung nun im folgenden Abschnitt beschrieben wird.

## 6.4 Erstellung von XML-Dateien mit Hilfe von Microsoft Access

Die Erstellung der XML-Datei erfolgt mit Hilfe von Makros und Abfragen in MS Access. Ziel dieses Schritts ist es, alle benötigten Informationen in einer, der oben beschriebenen DTD entsprechenden, XML-Datei darzustellen.

Auf die Analyse des Datenbestandes aufbauend können die nötigen Abfragen formuliert werden. Die Informationen über das Sigel, das Fachsemester, den Titel der Veranstaltung und die Anmerkungen zur Durchführung finden sich in der Tabelle `erfass`. Die Angabe des Moduls ist dort als Fremdschlüssel im Feld `fach` enthalten. Die Bezeichnung des Moduls kann durch Verknüpfung dieses Feldes mit dem Feld `ID` der Tabelle `modul` gewonnen werden. Die Angabe des Studiengangs erhält man durch die Verknüpfung mit der Tabelle `studiengang`. Die Durchführungsdauer und die SWS für die Studenten können durch die Verknüpfung über das Sigel mit der Tabelle `dozenten` gewonnen werden. Tag, Zeit, Raum und Gruppe erhält man aus der Tabelle `lv`. Die Verknüpfung findet auch hier über das Sigel statt. In der Tabelle `dozenten` ist statt des Dozentennamens dessen Nummer als Fremdschlüssel angegeben. Der Dozentename wird über die Verknüpfung dieses Feldes mit der Tabelle `dozadr` bezogen.

Wie schon ausgeführt wurde, können in einem Fach mehrere Gruppen in eigenen Veranstaltungen unterrichtet werden. Die Informationen dazu, wie zum Beispiel den Dozenten, finden sich in der Tabelle `lv`. In dieser Tabelle ist der Dozent im Feld `DozentG` jedoch nicht als Nummer, sondern unter dem Nachnamen eingetragen. Aus diesem Grund ist eine Ver-

knüpfung mit dem Feld `Name` der Tabelle `dozadr` notwendig, obwohl es sich im Falle des Felds `Name` nicht um den Primärschlüssel der Tabelle handelt. Zweckmäßiger wäre es gewesen, die Angabe des Dozenten in der Tabelle `lv` als Fremdschlüssel aus der Tabelle `dozadr` zu hinterlegen.

Die Folge dieser Art der Darstellung des Dozenten in der Tabelle `lv` ist auch, dass dadurch die Veranstaltungen, die von mehreren Dozenten gemeinsam geleitet werden, nicht in der Abfrage berücksichtigt werden können. Existierten mehrere Dozenten mit dem gleichen Nachnamen, wäre eine eindeutige Zuordnung ebenfalls nicht mehr möglich. Durch die Darstellung der Dozenten in einer weiteren Tabelle über die Felder (Dozenten-)Nummer (als Fremdschlüssel aus Tabelle `dozadr`), Siegel und (Gruppen-)Nummer (beide als Fremdschlüssel aus Tabelle `lv`) hätte die Zuordnung eindeutig dargestellt werden können.

Die Darstellung der Abfrage `plan_haupt` in Abbildung 8 zeigt die Tabellen, die zur Formulierung dieser Abfrage notwendig sind, deren selektierte Felder und die Verknüpfungen zwischen den Tabellen. Diese Abfrage wurde als Tabellenerstellungsabfrage erstellt, die die Tabelle `-plan` erzeugt<sup>37</sup>. An diese Tabelle müssen die anrechenbaren Veranstaltungen mit den jeweiligen individuellen Werten Studiengang, Semester und Modul angefügt werden. Dies ist durch eine Anfügeabfrage möglich und wird von der Abfrage `plan_anrechenbar` vorgenommen. Die anrechenbaren Veranstaltungen sind über die Felder `Siegel` und `Fach` (welches das Modul bezeichnet) näher spezifiziert. Das Feld `Fach` ist als Fremdschlüssel der Tabelle `modul` angegeben, über welche die Modulbezeichnung bezogen werden kann. Studiengang und Studiensemester sind ebenfalls aus der Tabelle `anrechenbarkeit` ersichtlich. Die weiteren Daten werden analog zur Abfrage `plan_haupt` ausgewählt (vgl. Abbildung 8).

Eine weitere Anfügeabfrage, die Abfrage `plan_fakultativ`, fügt die fakultativen Lehrveranstaltungen an die Tabelle `-plan` an. In dieser Abfrage werden nur diejenigen Einträge aus der Datenbank selektiert, bei denen der Feldwert des Feldes `Art` in der Tabelle `erfass` „Fakultative LV“ lautet. Auf die Felder `Studiengang` und `Semester` wird verzichtet, da die fakultativen Veranstaltungen von allen Studierenden gewählt werden können. Die weiteren Felder werden analog zur Abfrage `plan_haupt` ausgewählt (vgl. Abfrage `plan_fakultativ` in Abbildung 8)

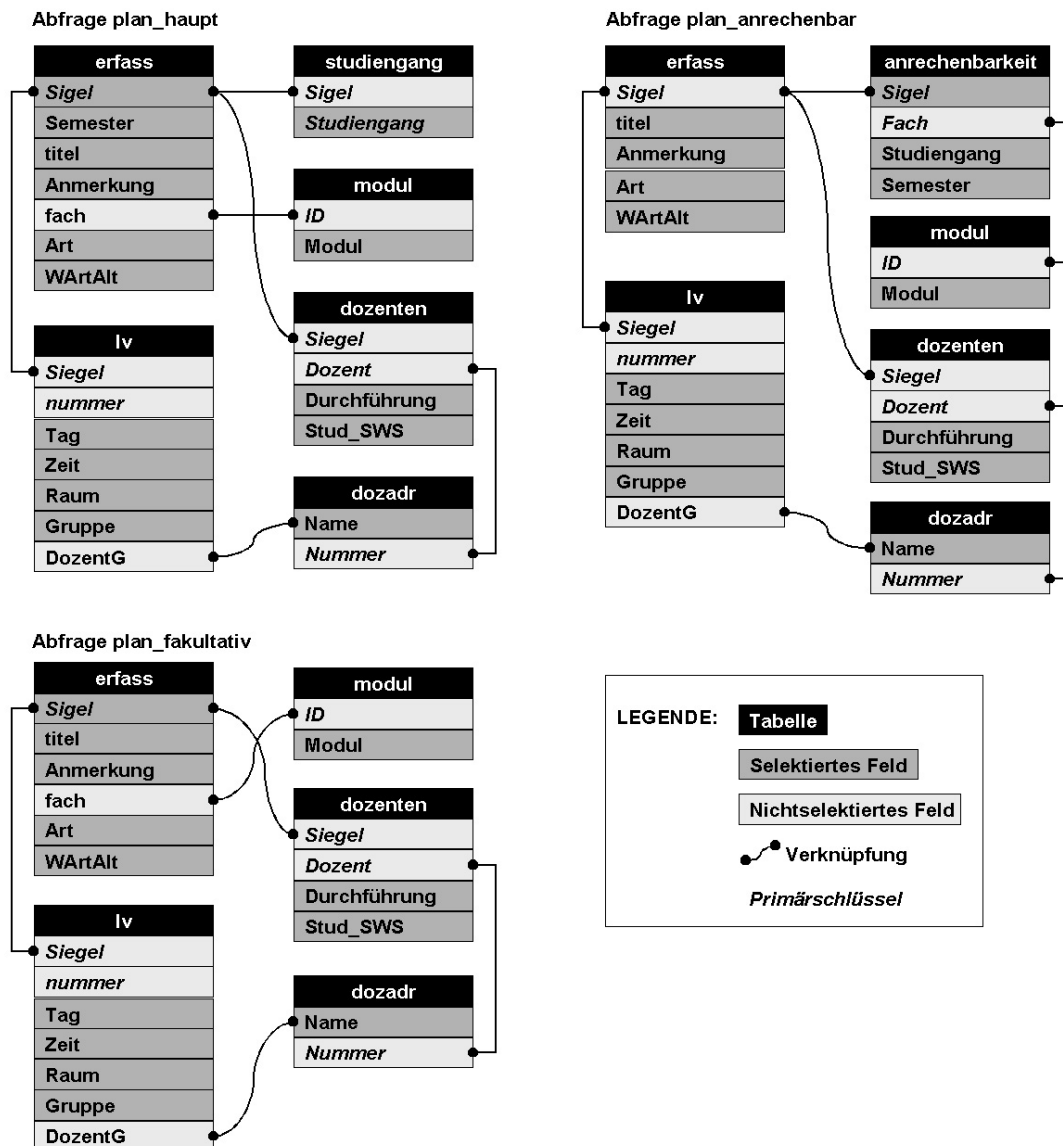


Abbildung 8: Abfragen zur Gewinnung der Daten für die XML-Darstellung des Stundenplans

Aus der dadurch gewonnenen Tabelle `-plan` wird die XML-Datei durch drei weitere Abfragen hergestellt.

XML-Dateien bestehen aus einer XML-Deklaration, der Referenz auf die DTD und aus den gemäß der DTD festgelegten Elementen. In der ersten Abfrage, wird die XML-

<sup>37</sup> Alle in der Datenbank formulierten Abfragen finden sich auf dem beigegeführten Datenträger (sie-

Deklaration, die Referenz auf die DTD und das Start-Tag des Wurzelements in eine neue Tabelle eingefügt. Dies geschieht mit Hilfe der Tabellenerstellungsabfrage `plan_XML-Start`, welche die Tabelle `-plan_export` erstellt. Da Abfragen jedoch nur über Tabellen vorgenommen werden können, muss an dieser Stelle ein Trick angewendet werden. In der Datenbank wurde zuvor die Tabelle `test` mit genau einem Datensatz angelegt und – um versehentliche Änderungen in dieser Tabelle zu vermeiden – aus der Datenblattansicht ausgeblendet. Über diese Tabelle wird die Abfrage ausgeführt, so dass der Eintrag genau einmal in die Tabelle `-plan_export` vorgenommen wird. Ebenfalls werden durch die Abfrage die Felder bestimmt, an die später die weiteren Datensätze angefügt werden.

Die SQL-Anweisung zu dieser Abfrage lautet:

```
SELECT DISTINCTROW "<?xml version='1.0' encoding='ISO-8859-1'?>" AS Sigel,
"<!DOCTYPE plan SYSTEM 'http://v.hdm-stuttgart.de/~spaleck/plan.dtd'" AS
Semester, "<plan>" AS Studiengang, "" AS Modul, "" AS titel, "" AS Name, "" AS
Durchführung, "" AS Stud_SWS, "" AS Anmerkung, "" AS Tag, "" AS Zeit, "" AS
Raum, "" AS Gruppe, "" AS Art, "" AS WArtAlt INTO [-plan_export] FROM test;
```

Durch die Anfügeabfrage `plan_export` werden nun die Datensätze der Tabelle `-plan` mit den entsprechenden XML-Tags versehen. Der Ausdrucks-Editor<sup>38</sup> ermöglicht die Verbindung der XML-Tags mit den Feldwerten. Zum Beispiel fügt der Ausdruck

```
"<studiengang>" & [-plan]![Studiengang] & "</studiengang>"
```

den Wert des Feldes `Studiengang` zwischen die entsprechenden Start- und Ende-Tags ein. Dem ersten Ausdruck der Abfrage ist zudem der Start-Tag des Elements `lv` voranzustellen, dem letzten Ausdruck der Abfrage ist das entsprechende Ende-Tag nachzustellen (vgl. Abbildung 9).

Eine weitere Anfügeabfrage, die Abfrage `plan_XML-Ende`, fügt entsprechend der Vorgehensweise bei der Abfrage `plan_XML-Start` das Ende-Tag des Wurzelements an. Die SQL-Anweisung lautet in diesem Fall:

```
INSERT INTO [-plan_export] ( Sigel, Semester, Studiengang, Modul, titel, Name,
Durchführung, Stud_SWS, Anmerkung, Tag, Zeit, Raum, Gruppe, Art, WArtAlt )
SELECT DISTINCTROW "</plan>" AS Sigel, "" AS Semester, "" AS Studiengang, ""
AS Modul, "" AS titel, "" AS Name, "" AS Durchführung, "" AS Stud_SWS, "" AS
Anmerkung, "" AS Tag, "" AS Zeit, "" AS Raum, "" AS Gruppe, "" AS Art, "" AS
WArtAlt
```

---

he *Anhang A: CD-ROM*)

<sup>38</sup> Der Ausdrucks-Editor wird über die Funktion „Aufbauen...“ im Kontextmenü an der Stelle, an der der Ausdruck eingefügt werden soll gestartet.

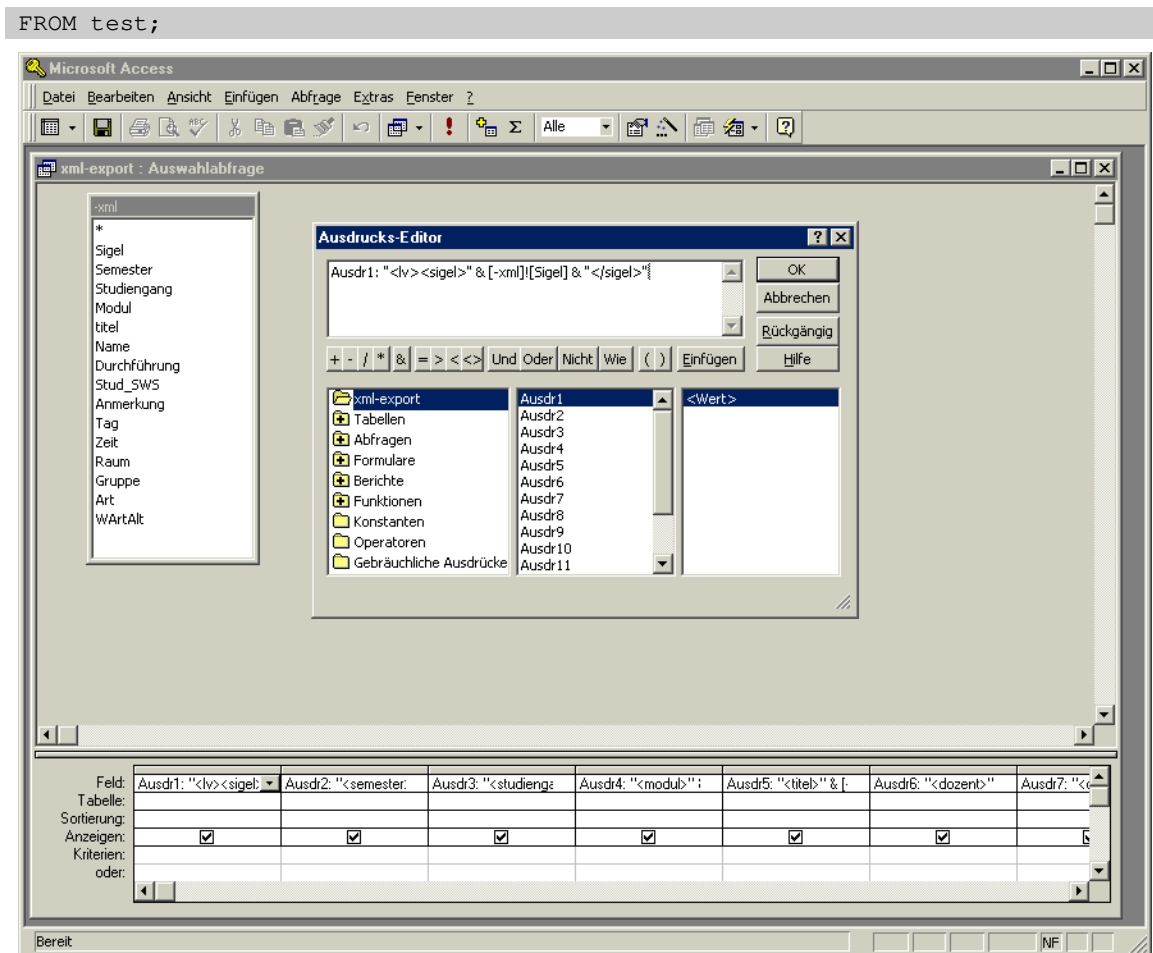


Abbildung 9: Formulierung von Abfragen mit Hilfe des Ausdrucks-Editors

Die dadurch gewonnene XML-Datei kann nun als XML-Dokument abgespeichert werden. Über die Funktion „Datei / *Speichern unter/Exportieren... / In eine externe Datei oder Datenbank*“ lässt sich das Exportfenster öffnen. Als Dateityp ist „Textdatei“ zu wählen, als Dateiname *plan.xml*. Über „*Exportieren*“ öffnet sich der Textexport-Assistent, bei dem die Funktion „Mit Trennzeichen...“ zu wählen ist. Im zweiten Schritt des Assistenten sind die Einstellungen „*Trennzeichen: Leerzeichen*“ und „*Textbegrenzungszeichen: {kein}*“ zu wählen. Mit „*Fertigstellen*“ wird der Export gestartet.

Die oben beschriebenen Schritte sind bis zum Schritt „*Speichern unter/Exportieren...*“ über das Makro *plan* realisiert.

Sollte der Export der XML-Datei aus der Datenbank öfter vorgenommen werden müssen, wäre es möglich diese Schritte mit Hilfe von Visual Basic Script weiter zu automatisieren.

Durch das nachträgliche Anfügen der anrechenbaren Veranstaltungen und der Verknüpfung zur Tabelle `semester` sind der XML-Datei zusätzliche Termine hinzugefügt worden, die sich in der Dozenten- und Raumauskunft negativ auswirken. Dadurch, dass ein Termin z.B. für mehrere Semester eingetragen wird, würde dieser bei der Raumauskunft mehrfach für den selben Zeitraum erscheinen. Dieses Problem entsteht auch bei der Dozentenanzeige.

Dieser negative Effekt lässt sich über eine zweite, kleinere XML-Datei beheben. Die Vorgehensweise ist bei dieser zweiten XML-Datei ähnlich der ersten. Über die Tabellenerstellungsabfrage `raum` werden die Datensätze für den XML-Export gemäß Abbildung 10 gewonnen. Bei dieser Abfrage wird auf die Verknüpfung zur Tabelle `semester` und die oben erläuterten Anfügeabfragen verzichtet. Die Informationen über den Studiengang und das Fachsemester werden bei dieser Abfrage aus der Tabelle `modul` gewonnen. Die Angabe der SWS kann unterbleiben, da sie nur für die studentischen Stundenpläne von Bedeutung sind.

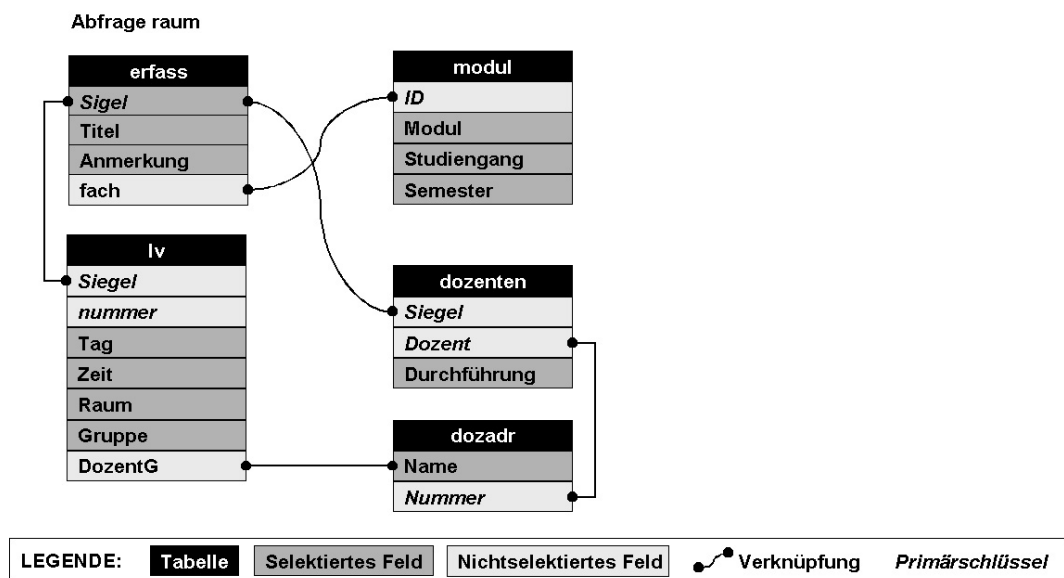


Abbildung 10: Abfrage zur Gewinnung der Daten für die XML-Darstellung der Dozenten- und Raumauskunft

Die Umwandlung der dadurch erstellten Tabelle `-raum` erfolgt nach dem gleichen Prinzip, wie der Erzeugung der XML-Datei `plan.xml`. Über die Tabellenerstellungsabfrage

raum\_XML-Start werden die XML-Deklaration, die Referenz zur DTD und das Start-Tag des Wurzelements in die Tabelle `-plan_export` eingefügt. Die DTD ist in diesem Fall die Datei *raum.dtd*, die sich nur unwesentlich von der in Abschnitt 6.3 dargestellten nicht-hierarchischen DTD unterscheidet:

```
<!ELEMENT plan (lv)+>
<!ELEMENT lv (sigel, modul, titel, dozent, durchfuehrung, anmerkung?, tag,
zeit, raum?, gruppe?, studiengang, semester)>
<!ELEMENT sigel (#PCDATA)>
<!ELEMENT modul (#PCDATA)>
<!ELEMENT titel (#PCDATA)>
<!ELEMENT dozent (#PCDATA)>
<!ELEMENT durchfuehrung (#PCDATA)>
<!ELEMENT anmerkung (#PCDATA)>
<!ELEMENT tag (#PCDATA)>
<!ELEMENT zeit (#PCDATA)>
<!ELEMENT raum (#PCDATA)>
<!ELEMENT gruppe (#PCDATA)>
<!ELEMENT studiengang (#PCDATA)>
<!ELEMENT semester (#PCDATA)>
```

Im nächsten Schritt werden die Datensätze der Tabelle `-raum` über die Anfügeabfrage `raum_export` mit den XML-Tags in die Tabelle `-raum_export` eingefügt. Das Ende-Tag des Wurzelements wird durch die Anfügeabfrage `raum_XML-Ende` angefügt. Dazu wurde analog zu der, bei der Abfrage `plan_XML-Ende` geschilderten Methode vorgegangen.

Beim Export dieser XML-Repräsentation ist wie bei der zuerst beschriebenen XML-Datei zu verfahren. Der Dateiname lautet in diesem Fall *raum.xml*. Das Ausführen der Abfragen, sowie das Öffnen des Exportfensters kann mit dem Makro `raum` abgekürzt werden. Beide XML-Dateien müssen schließlich noch mit FTP<sup>39</sup> auf dem Server bereitgestellt werden.

---

<sup>39</sup> FTP (File Transfer Protocol) regelt die Übertragung von Dateien über das Internet. FTP steht oft auch für Programme, die mit diesem Protokoll Daten senden und empfangen.

## 6.5 Erstellung von XSLT-Stylesheets zur Transformation in HTML

In diesem Abschnitt werden die Stylesheets vorgestellt, mit denen die Transformation der XML-Dateien vorgenommen wird. Zur Transformation dieser XML-Dateien wurden vier XSLT-Stylesheets entworfen:

- Das Stylesheet *plan.xsl* dient zur Erstellung der Stundenpläne der Studenten. Es arbeitet mit der XML-Datei *plan.xml*.
- Das Stylesheet *fakultativ.xsl* erstellt die Auflistung der fakultativen Lehrveranstaltungen. Zur Transformation verwendet es die XML-Datei *plan.xml*.
- Das Stylesheet *dozent.xsl* ist für die Transformation der XML-Datei *raum.xml* in Veranstaltungspläne für Dozenten zuständig.
- Die XSLT-Datei *raum.xsl* wird zur Transformation der XML-Datei *raum.xml* in Raumbelungspläne verwendet.

### 6.5.1 Gestalterische Aspekte

Die Stundenplananzeige soll für alle Ausgabearten einheitlich nach dem in Abbildung 11 dargestellten Schema aufgebaut sein.

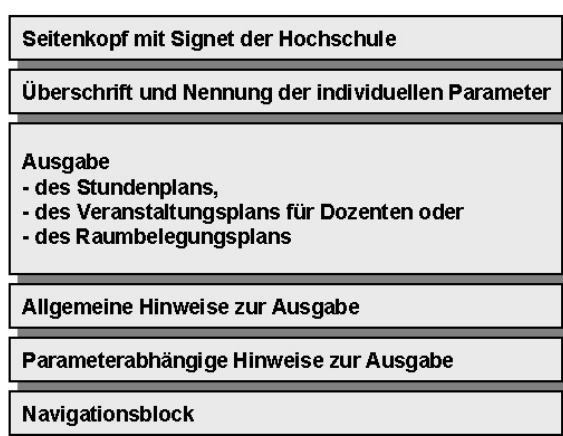


Abbildung 11: Struktureller Aufbau aller Ausgaben von Terminplänen



Die Ausgabe der Pläne beinhaltet jeweils einen Seitenkopf mit dem Signet der Hochschule, eine Überschrift mit der Nennung der jeweiligen benutzerabhängigen, individuellen Parameter, die Anzeige des Veranstaltungsplans, allgemeine und parameterabhängige Hinweise zur Ausgabe und einen Navigationsblock. Die allgemeinen Hinweise zur Ausgabe sollen das Ergebnis erläutern, die parameterabhängigen Hinweise sollen nutzerbezogene Erklärungen zur Ausgabe geben. Der Navigationsblock soll das Navigieren innerhalb des Auskunftssystems ermöglichen.

Sowohl im Hinblick auf die Anzeige der Pläne als auch in Bezug auf die Nutzersteuerung wurde versucht die Darstellung der Browseranzeigen dem Styleguide der Hochschule der Medien anzugleichen. Dies wurde insbesondere durch die Farbgestaltung, die der im Styleguide für den Fachbereich IuK festgelegten Farbgebung nahe kommt, realisiert. Bei der Ausgabe der Terminpläne wurde bewusst auf Hintergrundfarben und zu helle Schriftfarben verzichtet um ein Ausdrucken der Pläne in akzeptabler Qualität zu ermöglichen. Um die Einheitlichkeit der Formatierung von Textelementen zu gewährleisten wurden Cascading Style Sheets eingesetzt.

## 6.5.2 Stundenplanauskunft

Das Stylesheet, welches zur Erstellung der Stundenpläne der Studenten dient, besteht aus der Festlegung der Parameter sowie insgesamt sieben Templates. Der Dateiname des Stylesheets ist *plan.xsl*. Wie in Tabelle 1 dargestellt, können bis zu vier Module gewählt werden. Aus diesem Grund sind neben den Parametern *stg* (Studiengang) und *sem* (Fachsemester) die vier Parameter *wahl1*, *wahl2*, *wahl3* und *wahl4* vorgesehen. Diese enthalten die Modulbezeichnungen. Der folgende Code zeigt die XML-Deklaration zu Beginn der XSLT-Datei und das daran anschließende Stylesheet.

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:param name="stg"/>
<xsl:param name="sem"/>
<xsl:param name="wahl1"/>
<xsl:param name="wahl2"/>
<xsl:param name="wahl3"/>
```

```
<xsl:param name="wahl4"/>
```

```
[Template seite]40
```

```
[Template eintrag]
```

```
[Template tag]
```

```
[Template zeit]
```

```
[Template art]
```

```
[Template termin]
```

```
[Template raum]
```

```
</xsl:stylesheet>
```

In den Templates sind die Anweisungen zur Transformation in HTML enthalten. Das Template `seite` stellt das „Haupt-Template“ dar. Es enthält das HTML-Grundgerüst, bestehend aus dem HTML-Head, dem Seitenanfang und dem Seitenende. Aus diesem Template wird das Template `eintrag` aufgerufen. Der Seitenanfang besteht aus dem Signet der Hochschule, der Überschrift, der Anzeige der gewählten Parameter und der Tabellenüberschrift. Das Seitenende beinhaltet abschließende Bemerkungen zur Ausgabe und den Navigationsblock.

Eine Anforderung an das Auskunftssystem ist die chronologische Sortierung der Termine. Diese würde den Wochentag als erstes und die Uhrzeit als zweites Ordnungskriterium erfordern. Mit dem XSLT-Element `xsl:sort` lassen sich die Elemente des Context Nodes alphabetisch oder numerisch sortieren. Wie bereits dargestellt, ist der Wochentag in der Datenbank als nicht-numerisches Kürzel erfasst, was eine numerische Sortierung unmöglich macht und bei der alphabetischen Sortierung die Ordnungsreihenfolge Dienstag, Donnerstag, Freitag, Mittwoch, Montag, Samstag zur Folge hätte. Die Sortierung nach Wochentagen ist mit `xsl:sort` demnach nicht möglich. Daher wird das Template `eintrag`, welches die Listendarstellung der Termine erstellt, nacheinander für die Wochentage Montag bis Samstag, sowie für die möglichen Feldwerte<sup>41</sup> bei zeitlich noch nicht festgelegten Lehrveranstaltungen aufgerufen.

---

<sup>40</sup> An diesen Stellen sind Elemente der dargestellten Datei gekürzt. Die vollständigen Dateien sind auf dem beigefügten Datenträger enthalten (siehe *Anhang A: CD-ROM*)

<sup>41</sup> Zu Beginn der Arbeiten am beschriebenen System war ein leerer Feldwert im Feld `tag` Zeichen für noch nicht feststehende Lehrveranstaltungen. Während der Arbeiten wurden Änderungen an der Datenbank vorgenommen und der Feldwert für noch nicht feststehende Lehrveranstaltungen in „\$“ geändert. Um alle Möglichkeiten einzubeziehen wurden beide Alternativen bei der Erstellung der Stylesheets berücksichtigt.

Das Template endet mit allgemeinen und studiengangsbezogenen Hinweisen zur Ausgabe des Stundenplans sowie mit einem Navigationsblock, in dem zur Startseite der Stundenplanauskunft, zur Dozenten- und Raumauskunft und zur zuletzt angezeigten Seite verlinkt wird.

Der folgende Codeblock zeigt das Template `seite`. Zur besseren Orientierung sind die XSLT-Elemente in diesem und den nachfolgenden Codeblöcken fett dargestellt.

```
<xsl:template name="seite" match="/">
<html>
[... HTML-Code: HTML-Head und CSS-Einbindung ...]
<body>

<table border="0" width="780"><tr><td></td><td></td></tr></table>

<table border="0" width="780" cellpadding="0">
<tr><td><p class="plan">Stundenplan</p>
<p><b>Studiengang: </b><i><xsl:value-of select="$stg"/></i>
<br/><b>Semester: </b><i><xsl:value-of select="$sem"/></i>
<br/>
<xsl:choose>
<xsl:when test="$wahl1 or $wahl2 or $wahl3 or $wahl4"><b>Wahlmodule: </b>
<xsl:if test="$wahl1"><i><xsl:value-of select="$wahl1"/></i></xsl:if>
<xsl:if test="$wahl2"><i>, <xsl:value-of select="$wahl2"/></i></xsl:if>
<xsl:if test="$wahl3"><i>, <xsl:value-of select="$wahl3"/></i></xsl:if>
<xsl:if test="$wahl4"><i>, <xsl:value-of select="$wahl4"/></i></xsl:if>
</xsl:when>
</xsl:choose>
</p><br/></td></tr></table>

<table border="0" width="780" cellpadding="4">
<tr><th colspan="2">Tag / Zeit</th><th>Art</th><th>Veranstaltung
[Sigel]</th><th>Dozent/in</th><th>SWS</th><th>Raum</th></tr>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mo'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'di'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
```

```

</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mi'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'do'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'fr'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'sa'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="''"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'$'"/>
<xsl:with-param name="studiengang" select="$stg"/>
<xsl:with-param name="semester" select="$sem"/>
</xsl:call-template>
</table>
[... HTML-Code: Hinweise zur Ausgabe ...]
<xsl:if test="$stg='IM' or $stg='WB' or $stg='ÖB'">
[... HTML-Code: Hinweise für Studierende der Studiengänge IM, ÖB, WB ...]
</xsl:if>
<xsl:if test="$stg='BMD'">
[... HTML-Code: Hinweise für Studierende des Studiengangs BMD ...]
</xsl:if>
<xsl:if test="$stg='BMB'">

```

```
[... HTML-Code: Hinweise für Studierende des Studiengangs BMB ...]
</xsl:if>
[... HTML-Code: Navigationsblock ...]
</body>
</html>
</xsl:template>
```

Das Template `eintrag` ist für die Transformation in benutzerindividuelle Pläne verantwortlich. An dieser Stelle werden die vom Benutzer gewählten Parameterwerte zur Generierung der Ausgabe verarbeitet. Anhand eines XPath-Ausdrucks im XSLT-Element `xsl:for-each` werden im Quellbaum (der XML-Datei) die auf den XPath-Ausdruck zutreffenden Knoten selektiert. Die vollständige XSLT-Anweisung lautet:

```
<xsl:for-each select="plan/lv[tag=$wochentag]"> (...) </xsl:for-each>
```

Der Location Path dieser Anweisung gibt einen vom Context Node ausgehenden (hier relativen) Pfad an, der das Child Element `plan/lv` zur Verarbeitung auswählt. Dabei werden nur die Elemente selektiert, bei denen der Text Node des `tag`-Elements dem Wert des Parameters `$wochentag` entspricht. Der Context Node zu dieser Anweisung ist der Root Node, der zu Beginn des Templates `seite` im XSLT-Element `xsl:template` über das Attribut `match` angegeben wurde.

Im weiteren Verlauf des Templates wird mit `xsl:sort` die Sortierung nach Anfangszeit und Durchführungsdauer vorgenommen. Die Selektion der Termine findet weiter über eine `if`-Abfrage statt, bei der überprüft wird, ob die vorselektierten Termine auch auf die individuellen Parameterwerte des Benutzers zutreffen. Kontrolliert wird also neben dem Studiengang und dem Fachsemester auch, ob es sich um ein Pflichtfach (das in jedem Fall ausgegeben wird) oder eines der gewählten Module handelt. Die entsprechende XSLT-Anweisung lautet dem entsprechend:

```
<xsl:if test="(semester=$semester and studiengang=$studiengang) and
(art='Pflichtfach' or modul=$wahl1 or modul=$wahl2 or modul=$wahl3 or
modul=$wahl4)">
```

Wie bei der Analyse der Datenbank festgestellt wurde, sind für die Studiengänge BMB bzw. BMD neben den „eigenen“ Veranstaltungen, also solche für die Studiengänge BMB bzw. BMD selbst, auch die Termine mit dem Studiengangskürzel BM zu berücksichtigen. Aus der SPO wurde ermittelt, dass diese Studiengänge zwar einen Teil der Wahlfächer aus festgelegten Modulen wählen müssen, bei der Wahl der restlichen Fächer jedoch völlig frei sind (vgl. Tabelle 1, Seite 45). Eine Selektion nach gewählten Modulen kann daher bei die-

sen Studiengängen nicht stattfinden, da für sie alle im Bereich der Wahlmodule angebotenen Lehrveranstaltungen prinzipiell wählbar sind. Obige if-Abfrage muss daher in diesem Fall folgendermaßen abgewandelt werden:

```
<xsl:if test="semester=$semester and (studiengang=$studiengang or studiengang='BM') ">
```

Bei der Datenbankanalyse wurde zudem festgestellt, dass für die Masterstudiengänge alle Lehrveranstaltungen als Pflichtfächer angegeben sind. Um diesen Umstand zu berücksichtigen muss die if-Abfrage im Fall der Masterstudiengänge ebenfalls modifiziert werden. Sie lautet dann:

```
<xsl:if test="(semester=$semester and studiengang=$studiengang) and (modul=$wahl1 or modul=$wahl2 or modul=$wahl3 or modul=$wahl4) ">
```

Die Ausgabe des Stundenplans erfolgt tabellarisch in sieben Spalten mit den folgenden Einträgen: Tag, Zeit, Art, Modul, Titel, Sigel, Wahlpflichtart (für die Studiengänge IM, ÖB und WB), Gruppe, Anmerkung, Dozent, Semesterwochenstunden und Raum. Die Angaben über Modul, Titel, Sigel, Wahlpflichtart, Gruppe und Anmerkung werden als Block in einer Tabellenspalte dargestellt.

Das vollständige Template `eintrag` ist in nachstehendem Codeblock dargestellt:

```
<xsl:template name="eintrag">
  <xsl:param name="wochentag"/>
  <xsl:param name="studiengang"/>
  <xsl:param name="semester"/>
  <xsl:choose>
    <xsl:when test="$studiengang='BMB' or $studiengang='BMD'">
      <xsl:for-each select="plan/lv[tag=$wochentag]">
        <xsl:sort select="zeit" data-type="number"/>
        <xsl:sort select="durchfuehrung" data-type="number"/>
        <xsl:if test="semester=$semester and (studiengang=$studiengang or studiengang='BM') ">
          <tr><td valign="top">
            <xsl:call-template name="tag"/>
          </td><td valign="top" width="65">
            <xsl:call-template name="zeit"/>
          </td><td valign="top">
            <xsl:call-template name="art"/>
          </td><td valign="top">
            <xsl:call-template name="termin"/>
          </td><td valign="top">
```

```

<xsl:value-of select="dozent"/>
</td><td valign="top">
<xsl:value-of select="sws"/>
</td><td valign="top">
<xsl:call-template name="raum"/>
</td></tr>

<tr><td colspan="7"><hr color="#999999" size="1"/></td></tr>
</xsl:if>
</xsl:for-each>
</xsl:when>

<xsl:when test="$studiengang='Master IW' or $studiengang='Master BM'">
<xsl:for-each select="plan/lv[tag=$wochentag]">
<xsl:sort select="zeit" data-type="number"/>
<xsl:sort select="durchfuehrung" data-type="number"/>
<xsl:if test="(semester=$semester and studiengang=$studiengang) and
(modul=$wahl1 or modul=$wahl2 or modul=$wahl3 or modul=$wahl4)">
<tr><td valign="top">
<xsl:call-template name="tag"/>
</td><td valign="top" width="65">
<xsl:call-template name="zeit"/>
</td><td valign="top">
<xsl:call-template name="art"/>
</td><td valign="top">
<xsl:call-template name="termin"/>
</td><td valign="top">
<xsl:value-of select="dozent"/>
</td><td valign="top">
<xsl:value-of select="sws"/>
</td><td valign="top">
<xsl:call-template name="raum"/>
</td></tr>

<tr><td colspan="7"><hr color="#999999" size="1"/></td></tr>
</xsl:if>
</xsl:for-each>
</xsl:when>

<xsl:otherwise>
<xsl:for-each select="plan/lv[tag=$wochentag]">
<xsl:sort select="zeit" data-type="number"/>
<xsl:sort select="durchfuehrung" data-type="number"/>
<xsl:if test="(semester=$semester and studiengang=$studiengang) and
(art='Pflichtfach' or modul=$wahl1 or modul=$wahl2 or modul=$wahl3 or
modul=$wahl4)">

```

```

<tr><td valign="top">
<xsl:call-template name="tag"/>
</td><td valign="top" width="65">
<xsl:call-template name="zeit"/>
</td><td valign="top">
<xsl:call-template name="art"/>
</td><td valign="top">
<xsl:call-template name="termin"/>
</td><td valign="top">
<xsl:value-of select="dozent"/>
</td><td valign="top">
<xsl:value-of select="sws"/>
</td><td valign="top">
<xsl:call-template name="raum"/>
</td></tr>
<tr><td colspan="7"><hr color="#999999" size="1"/></td></tr>
</xsl:if>
</xsl:for-each>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

```

Der Eintrag des Tages, der Zeit, der Art, der Terminbeschreibung (Modul, Titel, Sigel, Gruppe, etc.) und des Raumes findet in eigenen Templates statt.

Die Darstellung der Durchführungszeit des Termins wird aus der Startzeit und der Durchführungsdauer gewonnen. Bestimmte Einträge wie die Angabe der Veranstaltungsart oder die Anmerkungen zu einer Lehrveranstaltung sind in der Ausgabe zur Hervorhebung farblich gekennzeichnet. Die Formatierung findet zu einem großen Teil mit CSS statt.

```

<xsl:template name="tag">
<xsl:choose>
<xsl:when test="tag!='' and tag!='$'">
<xsl:value-of select="tag"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</xsl:template>
<xsl:template name="zeit">
<xsl:choose>
<xsl:when test="zeit!='' and zeit!='0'">

```



```

<xsl:value-of select="7+zeit"/><xsl:value-of select="7+zeit+durchfuehrung"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</xsl:template>
<xsl:template name="art">
<xsl:choose>
<xsl:when test="art='Pflichtfach'"><font color="blue">p</font></xsl:when>
<xsl:when test="art='Wahlpflichtfach'"><font color="green">w</font></xsl:when>
<xsl:when test="art='Wahlpflichtbereich"42'"><font
color="green">w</font></xsl:when>
<xsl:when test="art='Wahlpflichtbereich'"><font
color="green">w</font></xsl:when>
</xsl:choose>
</xsl:template>
<xsl:template name="termin">
<div class="modul"><xsl:value-of select="modul"/>: </div><b> <xsl:value-of
select="titel"/></b> [<xsl:value-of select="sigel"/>] <xsl:if
test="wahlartalt!=''">/ (<xsl:value-of select="wahlartalt"/>)</xsl:if>
<xsl:if test="gruppe!=''"><div class="gruppe">Gruppe: <xsl:value-of
select="gruppe"/></div></xsl:if>
<xsl:if test="anmerkung!=''"><div class="anmerkung">Anmerkung: <xsl:value-of
select="anmerkung"/></div></xsl:if>
</xsl:template>
<xsl:template name="raum">
<xsl:choose>
<xsl:when test="raum!=''">
<xsl:value-of select="raum"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</xsl:template>

```

### 6.5.3 Anzeige fakultativer Lehrveranstaltungen

Zur Darstellung der fakultativen Lehrveranstaltungen ist keine Selektion nach benutzerindividuellen Parametern erforderlich, da diese Veranstaltungen für alle Studenten wählbar

---

<sup>42</sup> Mögliche Werte des Feldes art in der Stundenplandatenbank sind „Pflichtfach“, „Wahlpflichtfach“ und „Wahlpflichtbereich“. Um eine mögliche spätere Änderung dieses Schreibfehlers einzubeziehen, ist in diesem Stylesheet auch „Wahlpflichtbereich“ berücksichtigt.

sind. Die Selektion der Einträge findet nach der Art der Lehrveranstaltung statt. Die hier dargestellte XPath-Anweisung verdeutlicht dies:

```
<xsl:for-each select="plan/lv[tag=$wochentag and art='Fakultative LV']">
```

Ähnlich wie bei der in Abschnitt 6.5.2 beschriebenen Anweisung wird über den Location Path das Child Element `plan/lv` ausgewählt. Die Selektion innerhalb dieser Elemente findet nach dem `tag` und `art` statt. Das komplette Stylesheet sieht dementsprechend wie folgt aus:

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:template name="seite" match="/">
<html>
[... HTML-Code: HTML-Head and CSS-Einbindung ...]
<body>
<table border="0" width="780"><tr><td></td><td></td></tr></table>
<table border="0" width="780" cellpadding="0">
<tr><td><p class="plan">Fakultative Lehrveranstaltungen</p>
<br/></td></tr></table>
<table border="0" width="780" cellpadding="4">
<tr><th colspan="2">Tag / Zeit</th><th>Veranstaltung
[Sigel]</th><th>Dozent/in</th><th>SWS</th><th>Raum</th></tr>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mo'"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'di'"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mi'"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'do'"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'fr'"/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'sa'"/>
```

```

</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'"'/>
</xsl:call-template>
<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'$'"/>
</xsl:call-template>
</table>
[... HTML-Code: Allgemeine Hinweise zur Ausgabe ...]
[... HTML-Code: Navigationsblock ...]
</body>
</html>

</xsl:template>
<xsl:template name="eintrag">
<xsl:param name="wochentag"/>
<xsl:for-each select="plan/lv[tag=$wochentag and art='Fakultative LV']">
<xsl:sort select="zeit" data-type="number"/>
<xsl:sort select="durchfuehrung" data-type="number"/>
<tr><td valign="top">
<xsl:choose>
<xsl:when test="tag!='' and tag!='$'">
<xsl:value-of select="tag"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</td><td valign="top" width="65">
<xsl:choose>
<xsl:when test="zeit!='' and zeit!='0'">
<xsl:value-of select="7+zeit"/>-<xsl:value-of select="7+zeit+durchfuehrung"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</td><td valign="top">
<b><xsl:value-of select="titel"/></b> [<xsl:value-of select="sigel"/>]
<xsl:if test="gruppe!=''"><div class="gruppe">Gruppe: <xsl:value-of
select="gruppe"/></div></xsl:if>
<xsl:if test="anmerkung!=''"><div class="anmerkung">Anmerkung: <xsl:value-of
select="anmerkung"/></div></xsl:if>
</td><td valign="top">
<xsl:value-of select="dozent"/>
</td><td valign="top">

```

```

<xsl:value-of select="sws"/>
</td><td valign="top">
<xsl:choose>
<xsl:when test="raum!=''">
<xsl:value-of select="raum"/>
</xsl:when>
<xsl:otherwise>--</xsl:otherwise>
</xsl:choose>
</td></tr>
<tr><td colspan="7"><hr color="#999999" size="1"/></td></tr>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

#### 6.5.4 Dozentenauskunft

Die Generierung der Veranstaltungspläne für die Dozenten (XSLT-Stylesheet *dozenten.xsl*) folgt dem oben beschriebenen Schema. Der Transformationsprozess ist kürzer und benötigt weniger Anweisungen als bei der Ausgabe der Stundenpläne. Zur Transformation wird die XML-Datei *raum.xml* herangezogen. In der tabellarischen Ausgabe des Veranstaltungsplans erscheinen die Angaben über Tag, Zeit, Modul, Titel, Sigel, Gruppe, Anmerkungen, Studiengang, Studiensemester und Raum. Eine Veranstaltung kann für mehrere Semester, Studiengänge und/oder Module anrechenbar sein. In der Ausgabe werden die Angaben zum Studiensemester, Studiengang und Modul gemacht, für welche die Veranstaltung in erster Linie vorgesehen ist. Das XSLT-Stylesheet besteht aus den Templates *seite*, *eintrag*, *tag*, *zeit* und *termin*. Zur Verarbeitung ist nur ein benutzerindividueller Parameterwert, der Name des Dozenten, nötig. Das Stylesheet hat demnach folgende Struktur:

```

<?xml version='1.0' encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:param name="dozent"/>
[Template seite]
[Template eintrag]
[Template tag]
[Template zeit]
[Template termin]
</xsl:stylesheet>

```

Das Template `seite` enthält analog zur Stundenplanauskunft das HTML-Grundgerüst aus dem HTML-Head, Seitenanfang (Signet der Hochschule, Überschrift, Anzeige des Parameters und Tabellenüberschrift) und Seitenende (Anmerkungen zur Ausgabe und Navigationsblock). Dort wird die Sortierung nach Wochentagen vorgenommen und das Template `eintrag` aufgerufen:

```
<xsl:template name="seite" match="/">
<html>

[... HTML-Code: HTML-Head und CSS-Einbindung ...]

<body>

<table border="0" width="780"><tr><td></td><td></td></tr></table>

<table border="0" width="780" cellpadding="0">
<tr><td><p class="plan">Wochenplan</p>
<p><b>für Dozent/in </b><i><xsl:value-of select="$dozent"/></i></p>
<br/></td></tr></table>

<table border="0" width="800" cellpadding="4">

<tr><th colspan="2" width="110">Tag / Zeit</th><th>Veranstaltung
[Sigel]</th><th>Stg.</th><th>Sem.</th><th>Raum</th></tr>

<tr><td colspan="6"><hr color="#999999" size="1"/></td></tr>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mo'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'di'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'mi'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'do'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'fr'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="'sa'"/>
</xsl:call-template>

<xsl:call-template name="eintrag">
<xsl:with-param name="wochentag" select="''"/>
```

```

</xsl:call-template>
</table>
[... HTML-Code: Hinweise zur Ausgabe ...]
[... HTML-Code: Navigationsblock ...]
</body>
</html>
</xsl:template>

```

Im Template `eintrag` werden mit Hilfe eines XPath-Ausdrucks die auszugebenden Lehrveranstaltungen selektiert. Die Anweisung lautet:

```
<xsl:for-each select="plan/lv[tag=$wochentag and dozent=$dozent]">
```

Sie gibt, wie im vorangegangenen Stylesheet, einen Location Path an, der auch hier das Child Element `plan/lv` auswählt. Es werden jedoch nur die Elemente selektiert, bei denen der Text Node des Child Elements `tag` und der Text Node des Child Elements `dozent` mit dem jeweiligen Parameterwert übereinstimmen. Der Context Node zu dieser Anweisung ist auch hier der Root Node. Im Anschluss an die Auswahl der entsprechenden Knoten wird die Sortierung nach Anfangszeit vorgenommen. Die Daten zu den Lehrveranstaltungen werden tabellarisch dargestellt. Der nachstehende Codeblock zeigt das Template `eintrag`:

```

<xsl:template name="eintrag">
  <xsl:param name="wochentag"/>
  <xsl:for-each select="plan/lv[tag=$wochentag and dozent=$dozent]">
    <xsl:sort select="zeit" data-type="number"/>
    <tr><td valign="top">
      <xsl:call-template name="tag"/>
    </td><td valign="top">
      <xsl:call-template name="zeit"/>
    </td><td valign="top">
      <xsl:call-template name="termin"/>
    </td><td valign="top">
      <xsl:value-of select="studiengang"/>
    </td><td valign="top">
      <xsl:value-of select="semester"/>
    </td><td valign="top">
      <xsl:value-of select="raum"/>
    </td></tr>
    <tr><td colspan="6"><hr color="#999999" size="1"/></td></tr>
  </xsl:for-each>
</xsl:template>

```

Die Templates `tag`, `zeit` und `termin` setzen die jeweiligen Werte in die Ausgabe ein. Die Templates `tag` und `zeit` entsprechen denen des Stylesheets *plan.xml*. Im Template `termin` wird auf die Angabe der Wahlpflichtart der auslaufenden Diplomstudiengänge verzichtet, die für die Ausgabe nicht benötigt wird. Daher ist sie in der verarbeiteten XML-Datei auch nicht vorhanden.

### 6.5.5 Raumauskunft

Das Stylesheet *raum.xml* unterscheidet sich mit dem der Dozentenauskunft nur in kleinen Details. Auf eine ausführliche Darlegung des Codes wird daher verzichtet, lediglich die Unterschiede werden nachstehend dargelegt.

Wie das Stylesheet *dozent.xml*, arbeitet auch diese XSLT-Datei mit der XML-Datei *raum.xml*. Das Stylesheet enthält die Templates `seite`, `eintrag` und `termin`. Als verarbeitungsindividuelles Parameter wird das Parameter `raum` definiert.

Auf die Templates `zeit` und `tag` kann verzichtet werden. Eine Überprüfung, ob diese Elemente einen Wert enthalten kann unterbleiben, da in der Stundenplandatenbank für alle Vorlesungstermine, für die ein Raum festgelegt wurde, auch ein Tag und eine Uhrzeit bestimmt wurde.

Der XML-Quellbaum wird nach der folgenden XPath-Anweisung durchsucht:

```
<xsl:for-each select="plan/lv[tag=$wochentag and raum=$raum]">
```

Statt der Überprüfung anhand des Dozenten wie im zuvor beschriebenen Stylesheet wird hier nach einer Übereinstimmung mit dem Parameter `$raum` gesucht. Die Ausgabe enthält die Daten Tag, Zeit, Modul, Titel, Sigel, Gruppe, Anmerkung und Dozent.

## 6.6 Aufruf der Veranstaltungspläne im Netz

Im letzten Schritt der Erstellung des Terminauskunftssystems wurden die Dateien zum Aufruf des XSLT-Prozessors und zur Steuerung des Nutzers durch das System entworfen. Wie im Abschnitt 5.2.3 erläutert, wird zur Verarbeitung der Stylesheets der XSLT-Prozessor Sablotron und die XSLT-Erweiterung von PHP verwendet.

In den grundlegenden Anforderungen an das System wurde unter anderem gefordert, dass die Benutzung des Systems für den Nutzer möglichst einfach und selbsterklärend sein

sollte. Der Bearbeiter solle das System möglichst problemlos an die Anforderungen eines neuen Semesters anpassen können.

Um diesen Anforderungen gerecht zu werden, wurden für die Erstellung eines Veranstaltungsplans mehrere, vom Nutzer zu durchlaufende Schritte eingeplant. Der Nutzer soll somit durch das System geführt werden. Von einer zentralen Startseite gelangt er entweder zur Stundenplanauskunft oder zur Dozenten- und Raumauskunft. Die Arbeiten zur Erstellung der Nutzerführung und zum Aufruf des XSLT-Prozessors werden in den folgenden Abschnitten erläutert.

### 6.6.1 Nutzerführung und Erfassung der Parameterwerte

Zur Erstellung der individuellen Terminpläne sind mehrere Benutzereingaben erforderlich. Bei der Stundenplanauskunft ist die Auswahl des Studiengangs, des Fachsemesters und, von Fall zu Fall unterschiedlich, eine Auswahl von Modulen erforderlich. Die erforderliche Anzahl der auszuwählenden Parameter lässt sich aus Tabelle 1 (auf Seite 45) ableiten. In Tabelle 2 sind die jeweils notwendigen Parameterwerte, die vor dem Aufruf des XSLT-Prozessors erfasst werden müssen, dargestellt.

Tabelle 2: Zur Generierung des Stundenplans erforderliche Parameter

Studien- gang	Semester	Zur Generierung des Stundenplans erforderliche Parameter					
		stg	sem	wahl1	wahl2	wahl3	wahl4
IM, ÖB, WB	1	ja	ja	nein	nein	nein	nein
	2-7	ja	ja	ja	ja	ja	nein
IW	1-2	ja	ja	nein	nein	nein	nein
	3-7	ja	ja	ja	ja	nein	nein
ID	1-2	ja	ja	nein	nein	nein	nein
	3-7	ja	ja	ja	ja	nein	nein
BMB, BMD	1-7	ja	ja	nein	nein	nein	nein



Studiengang	Semester	Zur Generierung des Stundenplans erforderliche Parameter					
		stg	sem	wahl1	wahl2	wahl3	wahl4
Master BM	1-4	ja	ja	ja	ja	ja	ja
Master IW	1-2	ja	ja	ja	ja	ja	ja

Die Studiengänge BMB und BMD wählen ihre Wahlmodule aus dem selben Angebot, Studierende der Studiengänge IM, ÖB und WB wählen ihre Wahlmodule aus einem größtenteils gemeinsamen Angebot. Diese Studiengangsgruppen können somit weitgehend jeweils als eine Einheit betrachtet werden.

Die Datei *start.php* verarbeitet die Parameterangaben und überprüft sie auf Vollständigkeit. Sind die nötigen Angaben vorhanden, wird der XSLT-Prozessor aufgerufen und diesem die Parameterwerte übergeben.

Die erste Überprüfung untersucht die Parameter `$stud` und `$art`:

```
if ($stud!="IM" and $stud!="ÖB" and $stud!="WB" and $stud!="IW" and
    $stud!="BMB" and $stud!="BMD" and $stud!="ID" and $stud!="Master IW" and
    $stud!="Master BM" and $art!="fakultativ") {
    include('stud.php');
}
```

Der Parameter `$stud` bezeichnet das Studiengangskürzel. Für die Ausgabe des Plans der fakultativen Veranstaltungen wird der Parameter `$art` eingeführt. Hat dieser den Wert „fakultativ“, wird, wie unten gezeigt werden wird, die Übersicht über die fakultativen Lehrveranstaltungen ausgegeben.

Ist kein gültiger Studiengang angegeben und hat der Parameter `$art` nicht der Wert „fakultativ“, wird die Datei *stud.php* geladen, in der über ein Formular der Studiengang ausgewählt werden kann. Enthält der Parameter `$stud` einen gültigen Studiengang, wird überprüft, ob der Parameter `$sem`, der für das Fachsemester steht, einen Wert enthält:

```
elseif (($stud=="IM" or $stud=="ÖB" or $stud=="WB") and $sem=="") {
    include('diplomalt.php');
}
elseif ($stud=="IW" and $sem=="") {
    include('iwdiplom.php');
}
elseif (($stud=="BMB" or $stud=="BMD") and $sem=="") {
```

```

    include('bm.php');
}
elseif ($stud=="ID" and $sem=="") {
    include('id.php');
}
elseif ($stud=="Master IW" and $sem=="") {
    include('iwmaster.php');
}
elseif ($stud=="Master BM" and $sem=="") {
    include('bmmaster.php');
}

```

Ist kein Semester angegeben, wird eine jeweils dem gewählten Studiengang angepasste PHP-Datei geladen, die über ein Eingabeformular für das Fachsemester (Parameter `$sem`) und die evtl. anzugebenden Module (Parameter `$wa`, `$wb`, `$wc` und `$wd`) verfügt. Wurde ein Wert für den Parameter `$sem` angegeben, wird eine dritte Überprüfung hinsichtlich der Modulauswahl durchgeführt. Diese findet anhand der in Tabelle 2 dargestellten studienbezogenen Konstellationen statt. Überprüft wird, ob zumindest ein benötigter Parameter nicht mit einem Wert besetzt ist und ob ein Modul mehrfach ausgewählt wurde. Eingeschränkt wird die Überprüfung bezüglich der Semester, in denen von den Studierenden noch keine Module gewählt wurden:

```

elseif (($stud=="IM" or $stud=="ÖB" or $stud=="WB") and $sem!="1" and
(($wa=="" or $wb=="" or $wc=="") or ($wa==$wb or $wa==$wc or $wb==$wc))) {
    include('module.php');
}
elseif (($stud=="IW" or $stud=="ID") and (($wa=="" or $wb=="" or $wa==$wb) and
($sem!="1" and $sem!="2"))) {
    include('module.php');
}
elseif (($stud=="Master IW" or $stud=="Master BM") and $sem and (($wa=="" or
$wb=="" or $wc=="" or $wd=="") or ($wa==$wb or $wa==$wc or $wa==$wd or
$wb==$wc or $wb==$wd or $wc==$wd))) {
    include('module.php');
}

```

Trifft einer der hier dargestellten Fälle zu, wird die Datei *module.php* aufgerufen, durch die der Nutzer darauf hingewiesen wird, dass seine Modulauswahl fehlerhaft war. Verlaufen diese Überprüfungen alle negativ, bedeutet das, dass die für eine Verarbeitung benötigten Parameter vollständig vorhanden und die Voraussetzungen für einen Aufruf des XSLT-Prozessors gegeben sind:

```
elseif ($art=="fakultativ" and $stud==" " and $sem==" " and $wa==" " and $wb==" "
and $wc==" " and $wd==" "){
[... PHP-Code: Aufruf des XSLT-Prozessors zur Ausgabe der fak. Veranstaltungen...]
}
else {
[... PHP-Code: Aufruf des XSLT-Prozessors zur Ausgabe der Stundenpläne ...]
}
```

Im Fall der Dozenten- und Raumauskunft findet die Nutzersteuerung durch die PHP-Datei *dozent.php* statt. Diese PHP-Datei hat folgendes Aussehen:

```
<html>
<head>
<title>HdM Stundenplan</title>
<link rel=stylesheet type="text/css" href="style.css">
</head>
<body>
<?php
if ($name) {
[... PHP-Code: Aufruf des XSLT-Prozessors zur Ausgabe der Pläne für Dozenten ...]
}
elseif ($raum) {
[... PHP-Code: Aufruf des XSLT-Prozessors zur Ausgabe der Raumbesetzungspläne ...]
}
else {
echo "<table border=\"0\" width=\"780\"><tr><td><img src=\"img/hdmiuk.gif\"
align=\"left\"></td><td><img src=\"img/iuklogo.gif\"
align=\"right\"></td></tr></table>";
echo "<table border=\"0\" width=\"780\" cellpadding=\"15\">";
echo "<tr><td bgcolor=\"\#007F97\"><p class=\"h1\">Dozentenauskunft</p></td>";
echo "<td bgcolor=\"\#007F97\"><p
class=\"h1\">Raumauskunft</p></td></tr><tr><td bgcolor=\"\#A8D3DA\"
valign=\"top\">";
include ("dozauswahl.php");
echo "</td><td bgcolor=\"\#A8D3DA\" width=\"50%\" valign=\"top\">";
include ("raumauswahl.php");
echo "</td></tr>";
echo "<tr><td colspan=\"2\" bgcolor=\"\#A8D3DA\"> <a href=\"start.php\"><img
src=\"img/pfeil.gif\" border=\"0\"> zur Stundenplanauskunft</a>";
echo "</td></tr></table>";
}
?>
```

```
</body>
</html>
```

Wie aus obigem Codeblock ersichtlich ist, findet die Überprüfung dahingehend statt, ob einer der Parameter `$name` oder `$raum` angegeben wurde. Ist keiner dieser Parameter belegt, wird das Eingabeformular zur Erfassung des Parameterwerts angezeigt. Die beiden Eingabeformulare werden jeweils über die Include-Funktion aus einer gesonderten PHP-Datei eingefügt. Ist einer der Parameter `$name` oder `$raum` angegeben, wird entsprechend der XSLT-Prozessor zur Verarbeitung aufgerufen.

### 6.6.2 Aufruf des XSLT-Prozessors

Nachdem die jeweils notwendigen Parameter erfasst wurden, wird der XSLT-Prozessor innerhalb des PHP-Skripts gestartet. Der folgenden Codeblock zeigt den Aufruf des XSLT-Prozessors am Beispiel der Stundenplanauskunft:

```
$processor = xslt_create();
if ($stud) {
    $xslt_params["stg"] = utf8_encode ($stud . "\x00");
}
if ($sem) {
    $xslt_params["sem"] = utf8_encode ($sem . "\x00");
}
if ($wa) {
    $xslt_params["wahl1"] = utf8_encode ($wa . "\x00");
}
if ($wb) {
    $xslt_params["wahl2"] = utf8_encode ($wb . "\x00");
}
if ($wc) {
    $xslt_params["wahl3"] = utf8_encode ($wc . "\x00");
}
if ($wd) {
    $xslt_params["wahl4"] = utf8_encode ($wd . "\x00");
}
xslt_run($processor, "/home/spaleck/public_html/plan.xsl",
"/home/spaleck/public_html/plan.xml", "arg:/_result", $xslt_params);
$result = xslt_fetch_result ($processor);
echo $result;
xslt_free($processor);
```

Zu Beginn wird ein neuer XSLT-Prozessor erzeugt. Im Anschluss daran werden die besetzten Parameter in UTF8-Codierung in den Hash `$xslt_params` geschrieben. Dazu wird den Parameterwerten die Zeichenkette „\x00“, eine Null in ASCII-Codierung, angefügt. Diese Zeichenkette wird zur Markierung einer Datenstruktur bei der Datenübertragung verwendet. Beim Weglassen der Zeichenkette wird der Parameterwert teilweise nicht korrekt verarbeitet und das Ergebnis in der Ausgabe somit verfälscht. Dieser nicht reproduzierbare Fehler steht möglicherweise im Zusammenhang mit dem bislang noch experimentellen Charakter der verwendeten PHP-Erweiterung (vgl. Abschnitt 5.2.3). Mit dem Anfügen der Zeichenkette „\x00“ können die Grenzen der Datenstruktur gekennzeichnet und vom verarbeitenden XSLT-Prozessor erkannt werden.

Über die Funktion `xslt_run` wird der Prozessor `$processor` aufgerufen und dem Stylesheet die XML-Datei sowie die Parameter zugewiesen. Über die Funktionen

```
$result = xslt_fetch_result ($processor);  
echo $result;
```

wird das Ergebnis der XSLT-Verarbeitung abgerufen und ausgegeben. Durch die Funktion `xslt_free` wird der XSLT-Prozessor wieder freigegeben.

Dieser Vorgang läuft bei der Erstellung der Veranstaltungspläne der Dozenten, der Raumbelegungspläne und der Übersicht über die fakultativen Lehrveranstaltungen analog ab.

## 7 Funktionsweise des Systems

Die Funktionsweise des realisierten Prototyps wird in den folgenden Abschnitten erläutert. Dabei wird zunächst auf die Funktionsweise aus der Sicht des Bearbeiters eingegangen, dessen Aufgabe die Pflege des System ist. Im Anschluss daran wird dargestellt, wie die Stundenplanauskunft aus der Benutzersicht arbeitet.

### 7.1 Funktionsweise aus der Sicht des Bearbeiters

Die Aufgabe des Bearbeiters umfasst zwei unterschiedliche Bereiche:

- die Aktualisierung der XML-Repräsentanz des Stundenplans und
- die Anpassung der Nutzersteuerung an die Gegebenheiten eines neuen Semesters.

Beide Aufgabenbereiche sind voneinander getrennt und erfordern verschiedene Kenntnisse. Es ist daher nicht zwingend notwendig, dass beide Arbeiten von der selben Person durchgeführt werden. Die erste Aufgabe fällt auf Grund der Tätigkeiten eher in den Aufgabenbereich des Stundenplanamts, die zweite Aufgabe könnte sowohl vom Stundenplanamt als auch von Angestellten im technischen Bereich durchgeführt werden.

#### 7.1.1 Aktualisierung der XML-Repräsentanz des Stundenplans

Die Aktualisierung der XML-Repräsentanz besteht aus drei Schritten: dem Import der Datenquelle, der Generierung der XML-Dateien und dem Kopieren dieser ins Netz. Für diese Aufgaben werden grundlegende Kenntnisse in Microsoft Access und in der Bedienung eines FTP-Clients (z.B.: WS\_FTP<sup>43</sup>) benötigt.

Grundlage für das Importieren der Datenquelle ist die Konfiguration des MyODBC-Treibers. Die Vorgehensweise bei der Konfiguration und beim Import in MS Access wurde in Abschnitt 6.2.1 bereits ausführlich beschrieben. Zu beachten gilt dabei allerdings, dass bei einem erneuten Import der Datenquelle in die Access-Datenbank, die zu ersetzenden

---

<sup>43</sup> WS\_FTP ist ein File Transfer Client der Firma Ipswitch, Inc.

Tabellen zunächst gelöscht werden müssen. Andernfalls würden die Tabellen mit falschen Tabellenbezeichnungen importiert werden.

Die weiteren Arbeiten in MS Access beschränken sich dann auf das Ausführen der beiden Makros `plan` und `raum` und dem anschließenden exportieren der jeweiligen Ergebnisse als XML-Dateien.

Im Anschluss daran müssen die exportierten XML-Dateien mit einen FTP-Client zur Verarbeitung ins Netz gestellt werden. Über den FTP-Client wird eine Verbindung zum Online-Laufwerk hergestellt, in das die Dateien kopiert werden müssen. Den Kopiervorgang zeigt Abbildung 12.

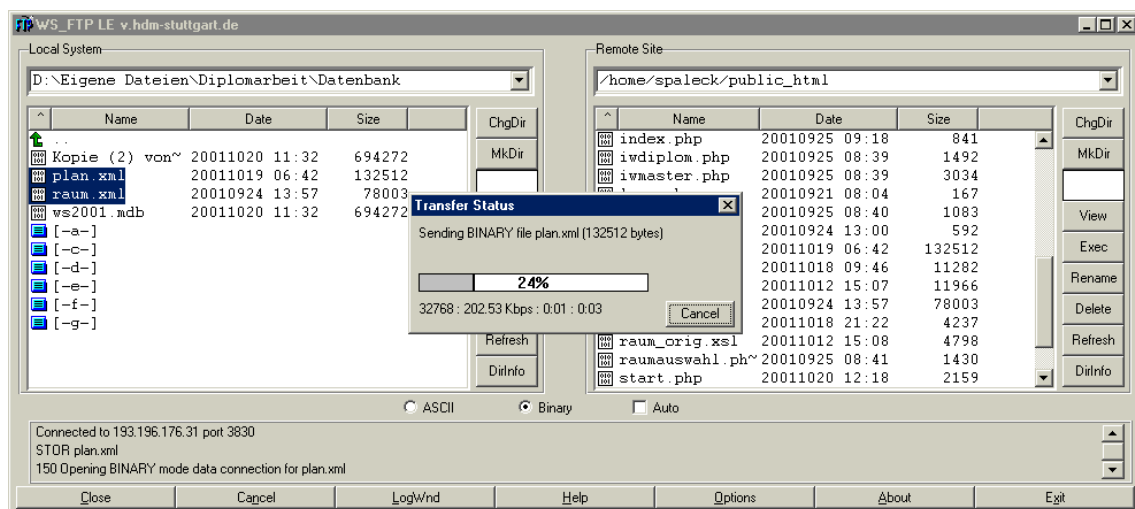


Abbildung 12: Kopiervorgang mit WS\_FTP

### 7.1.2 Anpassung der Nutzersteuerung

Zur Anpassung der Nutzersteuerung werden Grundlagenkenntnisse in HTML und PHP vorausgesetzt. Die Aufgabe besteht darin, die Eingabeformulare den Gegebenheiten eines neuen Semesters anzupassen. Dabei handelt es sich bei der Stundenplanauskunft in erster Linie um die Anpassung der Semestereingabe und um eine mögliche Änderung der Benennungen der Wahlmodule. Der folgende Codeblock zeigt die (möglicherweise) zu ändernden Einträge am Beispiel der Datei *diplomalt.php* fett hervorgehoben:

- Im oberen Teil der Datei die Angabe der möglichen Studiensemester im betreffenden Studiengang

- Im mittleren Teil der Datei die wählbaren Module (in der Schreibweise, die in der Stundenplandatenbank verwendet wurde)

[illegible]



```
Dozenten- und Raumauskunft</a>
<br> <a href="javascript:history.back()">
zurück</a></td></tr>
</table>
```

Diese Änderungen sind an den in Tabelle 3 dargestellten Dateien analog durchzuführen.

Tabelle 3: Durchzuführende Anpassungen in der Stundenplanauskunft

Studiengang	Datei	Änderung von...
IM, ÖB, WB	<i>diplomalt.php</i>	Semester, evtl. Module
IW	<i>indiplom.php</i>	Semester, evtl. Module
ID	<i>id.php</i>	Semester, evtl. Module
BMB, BMD	<i>bm.php</i>	Semester
Master BM	<i>bmmaster.php</i>	Semester, evtl. Module
Master IW	<i>invmaster.php</i>	Semester, evtl. Module

Die Anpassungen in der Dozenten- und Raumauskunft betreffen insbesondere die Angaben in der Datei *dozentenwahl.php*. Diese enthält die Namen der Lehrkräfte, die dem Benutzer zur Auswahl angeboten werden. Änderungen sind vorzunehmen, wenn neue Dozenten hinzukommen bzw. bestehende Einträge wegfallen.

Die Datei *raumauswahl.php* umfasst die Auswahl der Raumbezeichnungen, für die eine Raumauskunft möglich ist. Hier sind nur Änderungen vorzunehmen, wenn sich Raumbezeichnungen ändern, Hörsäle hinzukommen oder wegfallen.

## 7.2 Funktionsweise aus der Sicht des Benutzers

In diesem Abschnitt wird die Funktionsweise des Systems aus der Sicht des Benutzers erläutert. Der Benutzer gelangt über die Eingangsseite<sup>44</sup> auf das Auskunftssystem (Abbildung 13). Von dort aus hat er die Möglichkeit die Stundenplanauskunft, die Dozentenauskunft oder die Raumauskunft aufzurufen.

<sup>44</sup> <http://v.bdm-stuttgart.de/~spaleck>

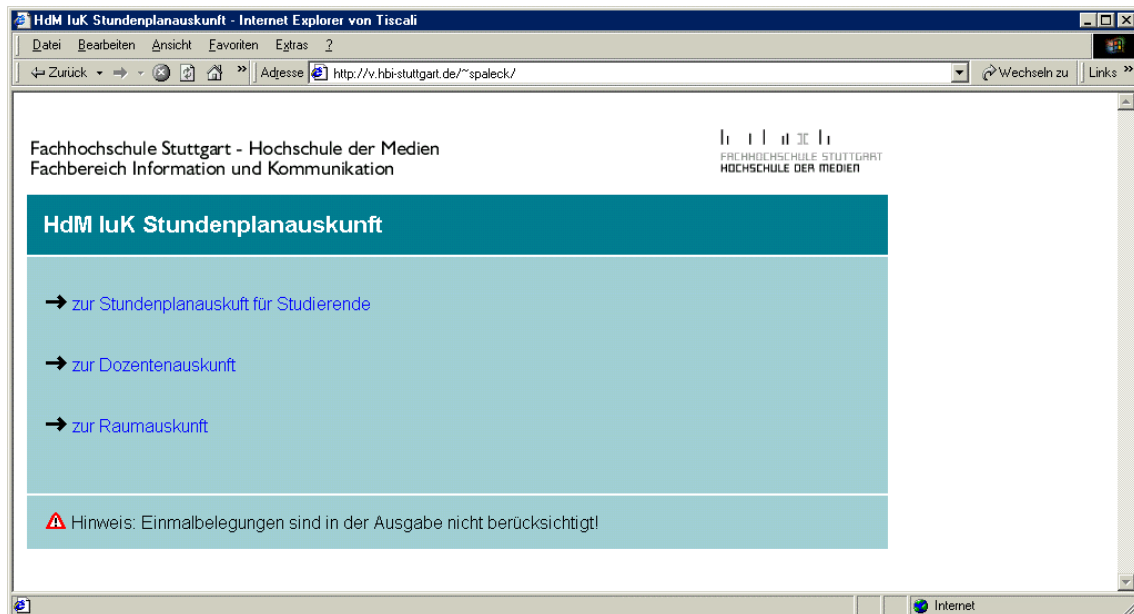


Abbildung 13: Eingangsseite des Auskunftssystems

### 7.2.1 Stundenplanauskunft für Studierende

Auf der Eingangsseite der Stundenplanauskunft wird der Benutzer aufgefordert einen Studiengang auszuwählen. Die Auswahl zeigt alle derzeit im Fachbereich IuK durchgeführten Studiengänge (Abbildung 14). Zur weiteren Auswahl des Semesters und Wahlmodule gelangt der Benutzer nur durch die Wahl eines gültigen Studiengangs. Im unteren Teil der Seite besteht die Möglichkeit, sich die fakultativen Lehrveranstaltungen anzeigen zu lassen (Abbildung 15). Darüber hinaus befindet sich am Seitenende auf allen Auswahlseiten der Link zur Dozenten- und Raumankunft.

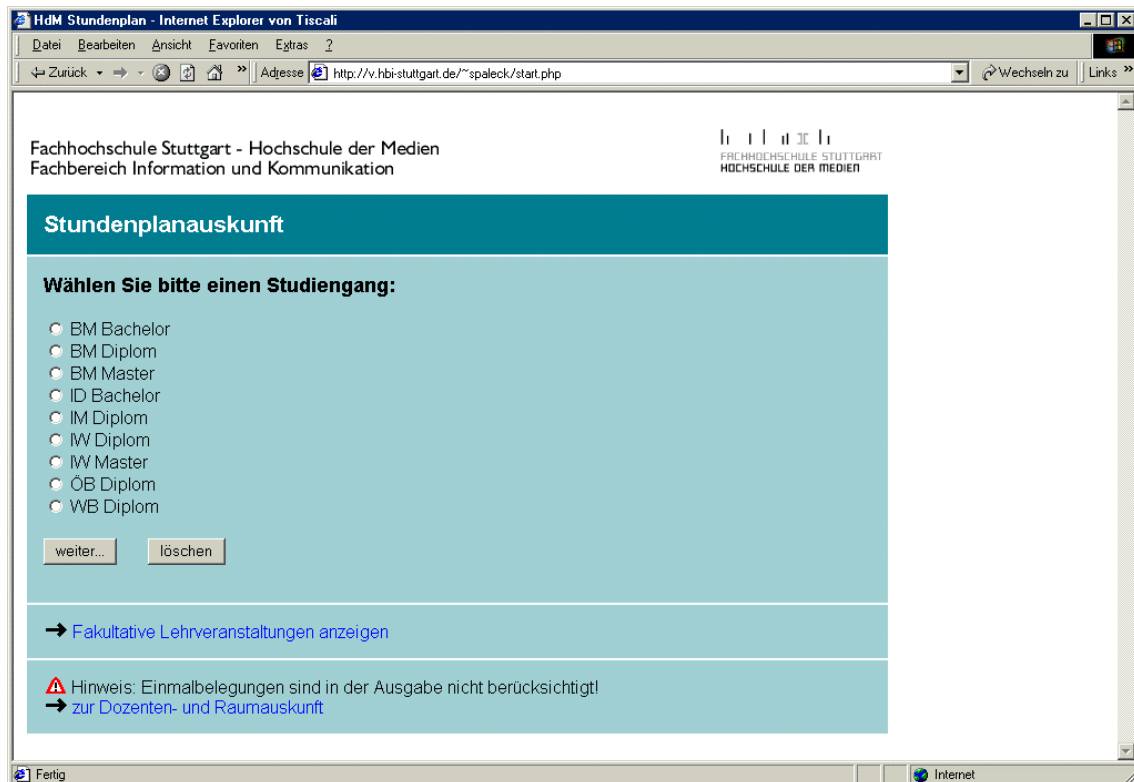


Abbildung 14: Stundenplanauskunft: Auswahl des Studiengangs

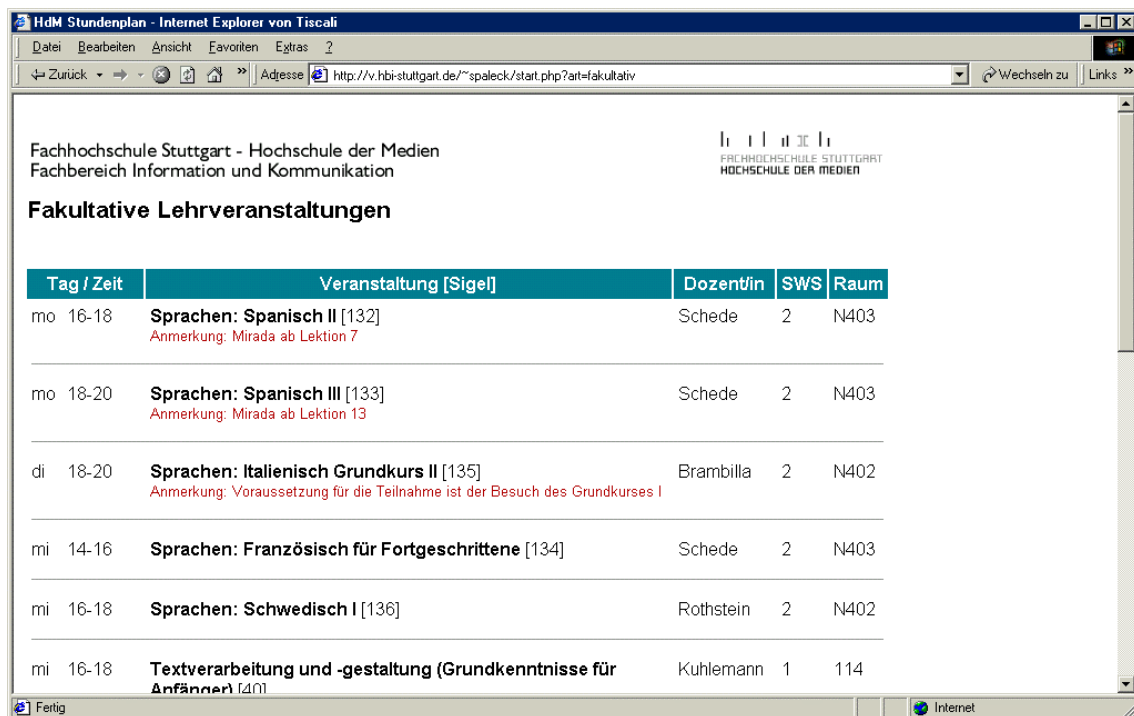


Abbildung 15: Stundenplanauskunft: Anzeige der fakultativen Veranstaltungen

Nach der Wahl des Studiengangs hat der Benutzer Eingaben zum Studiensemester und zu den gewählten Wahlmodulen zu treffen. Die Anzeige richtet sich nach dem zuvor ausgewählten Studiengang, d.h. je nach Auswahl des Studiengangs erhält der Benutzer eine andere, auf den individuellen Fall abgestimmte Anzeige. Abbildung 16 verdeutlicht die Auswahl des Fachsemesters und der Module am Beispiel des Studiengangs IM. In diesem konkreten Fall besteht für den Benutzer lediglich die Möglichkeit das 7. Semester auszuwählen, da im Stundenplansystem für diesen Studiengang nur Veranstaltungen dieses Semesters vorhanden sind. Die Auswahl der Wahlmodule erfolgt aus den in den verschiedenen Auswahllisten angebotenen Modulen.

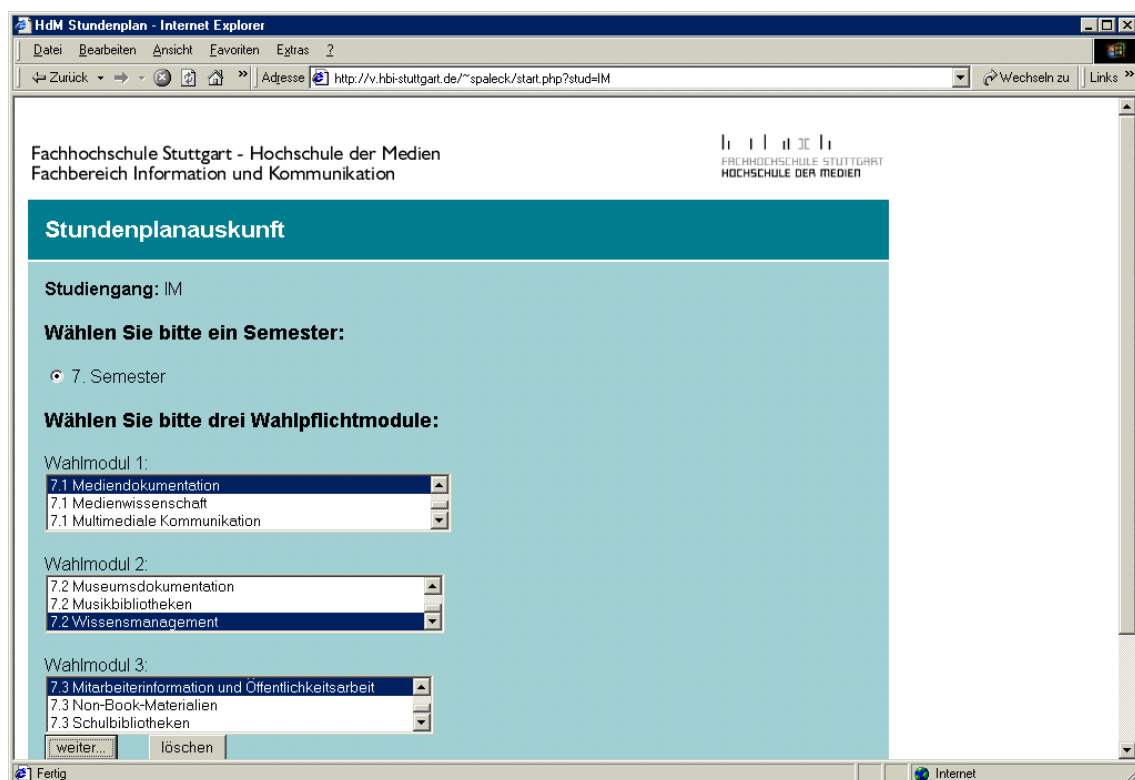


Abbildung 16: Stundenplanauskunft: Auswahl des Semesters und der Module

Sofern eine gültige Auswahl getroffen wurde, erhält der Benutzer den gemäß seinen Eingaben zusammengestellten Stundenplan. Abbildung 17 zeigt eine beispielhafte, individuelle Ausgabe eines Stundenplans. Zur Bestätigung der Eingaben werden vor dem Stundenplan nochmals die erfassten Parameter angezeigt.

Fachhochschule Stuttgart - Hochschule der Medien  
Fachbereich Information und Kommunikation

**Stundenplan**

Studiengang: IM  
Semester: 7  
Wahlmodule: 7.1 Mediendokumentation, 7.2 Wissensmanagement, 7.3 Mitarbeiterinformation und Öffentlichkeitsarbeit

Tag / Zeit	Art	Veranstaltung [Sigel]	Dozent/in	SWS	Raum
mo 10-12	w	7.3 Mitarbeiterinformation und Öffentlichkeitsarbeit: <b>Wissenschaftliche Aspekte der Unternehmens-PR</b> [129] / (pflicht)	Grudowski	2	N402
mo 16-18	w	7.3 Mitarbeiterinformation und Öffentlichkeitsarbeit: <b>Rhetorik und Chart-Technik der Management- und Business-Präsentation</b> [130] / (wahlfrei) <i>Anmerkung: Kenntnisse des Programms MS Power Point werden vorausgesetzt</i>	Grudowski	2	113
di 12-14	w	7.1 Mediendokumentation: <b>Drehbuchumsetzung</b> [154] / (pflicht)	Ehrmann	2	317
di 14-16	w	7.1 Mediendokumentation: <b>Drehbuchumsetzung</b> [154] / (pflicht)	Speck	2	317

Abbildung 17: Stundenplanauskunft: Ausgabe (oberer Teil)

-- -- p Dokumenten- und Workflowmanagement:  
**Dokumenten- und Workflowmanagement II** [124]  
*Anmerkung: teilweise geblockt, siehe Aushang*

**Allgemeine Hinweise:**

- ⚠ Die angeführten Veranstaltungen beginnen in aller Regel c.t. (sofern nicht anders in den Anmerkungen angegeben).
- ⚠ p = Pflichtfach; w = Wahlfach/Wahlpflichtfach

**Hinweise für die Studiengänge IM, ÖB und WB:**

- ⚠ Zu belegen sind in gewählten Wahlpflichtbereichen die mit *p* gekennzeichneten Veranstaltungen. Insgesamt müssen während des gesamten Studiums im Bereich 7.1 16 SWS, im Bereich 7.2 12 SWS und im Bereich 7.3 8 SWS belegt werden.

→ [zur Stundenplanauskunft](#)  
→ [zur Dozenten- und Raumauskunft](#)  
← [zurück](#)

Abbildung 18: Stundenplanauskunft: Ausgabe (unterer Teil)

Im unteren Teil der Ausgabe (Abbildung 18) werden, im Anschluss an die Stundenplanda-  
ten, Hinweise zur Ausgabe angezeigt. Bei der Anzeige von Plänen der Studiengänge IM,

ÖB, WB, BMB und BMD werden zusätzliche Anmerkungen zur Wahl der angezeigten Wahlmodule ausgegeben. Das letzte Element der Ausgabe bildet der Navigationsblock.

Für den Fall, dass der Benutzer bei der Auswahl der Module ungültige Angaben gemacht hat, wird kein Stundenplan ausgegeben. Stattdessen wird der Benutzer dazu aufgefordert seine Eingaben zu korrigieren. Als ungültige Eingaben werden angesehen:

- die mehrfache Auswahl eines Wahlmoduls oder/und
- eine unzureichende Anzahl angegebener Wahlmodule.

Diese Überprüfung findet nur statt, wenn der Benutzer ein Studiensemester ausgewählt hat, in welchem Wahlmodule vorgesehen sind.

## 7.2.2 Dozentenauskunft

Die Dozentenauskunft ermöglicht die Ausgabe von Veranstaltungsplänen für Lehrkräfte im Fachbereich IuK. Die Anzeige der Pläne erfolgt gemäß der durch den Benutzer im Bereich Dozentenauskunft getroffenen Auswahl (Abbildung 19).

The screenshot shows a web browser window titled 'HdM Stundenplan - Internet Explorer'. The address bar shows 'http://v.hbi-stuttgart.de/~spaleck/dozent.php'. The page header includes 'Fachhochschule Stuttgart - Hochschule der Medien' and 'Fachbereich Information und Kommunikation'. The main content area is split into two columns. The left column, 'Dozentenauskunft', has a heading 'Wählen Sie bitte:' followed by a scrollable list of names and a 'weiter...' button. The right column, 'Raumauskunft', has a heading 'Wählen Sie bitte:' and four sub-sections for different buildings and floors, each with radio buttons for selection and a 'weiter...' button. At the bottom, there is a link 'zur Stundenplanauskunft' and a 'Fertig' button.

Abbildung 19: Dozenten- und Raumauskunft

Die Ausgabe der Wochenpläne der Dozenten ist ähnlich strukturiert wie die Stundenplananzeige der Studierenden: zur Bestätigung wird vor der Anzeige des Wochenplans der gewählte Parameter (in diesem Fall der Name der Lehrkraft) angezeigt. Im Anschluss an den Veranstaltungsplan folgen allgemeine Hinweise und der Navigationsblock. (Abbildung 20).

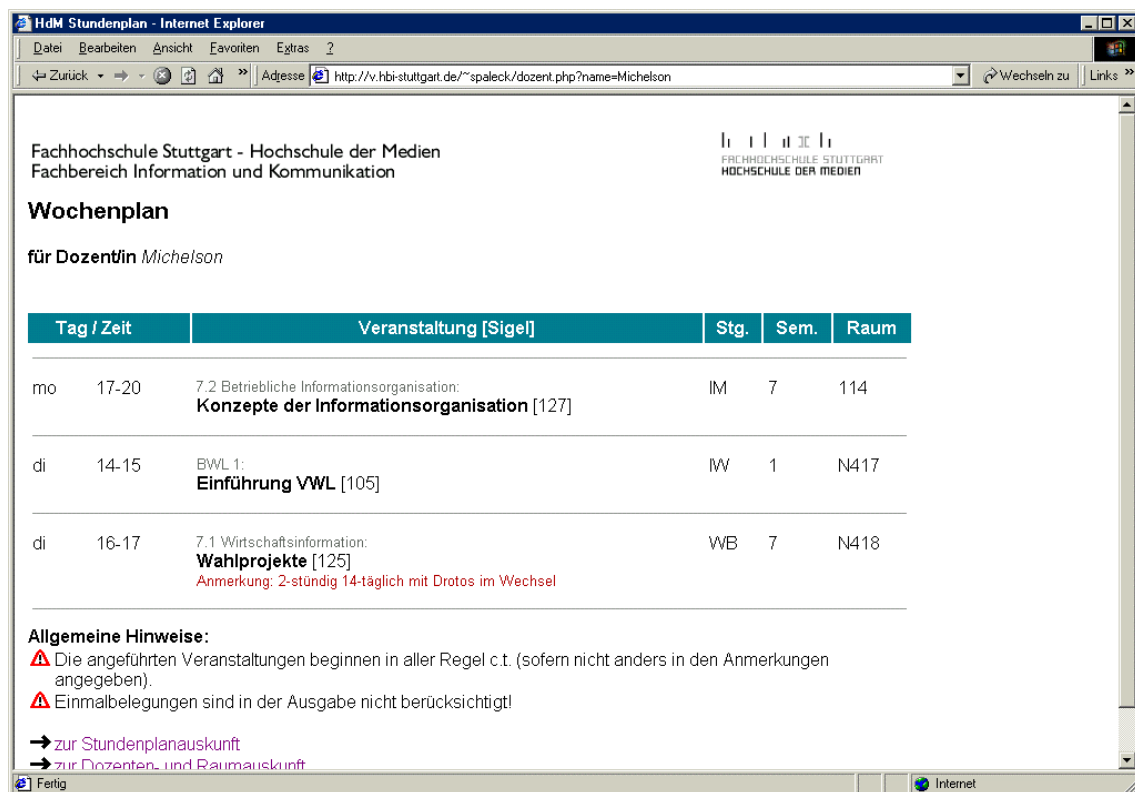
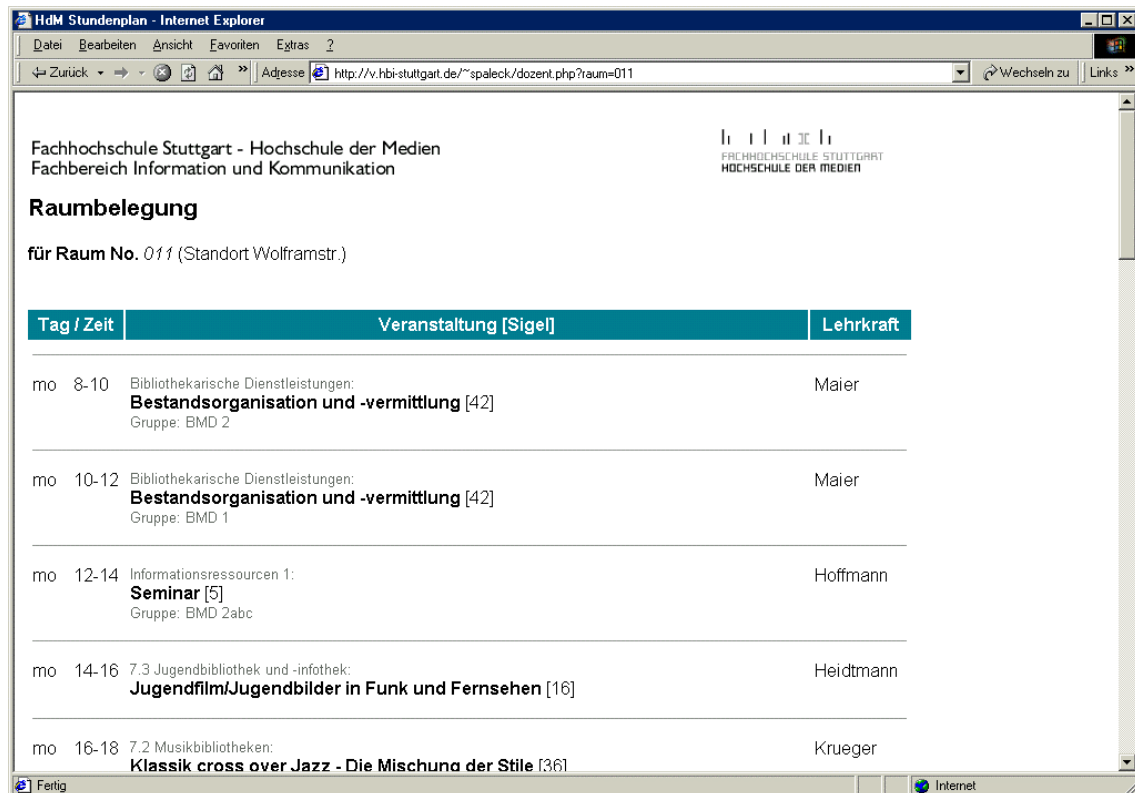


Abbildung 20: Dozentenauskunft: Ausgabe

### 7.2.3 Raumauskunft

Die Raumauskunft ermöglicht die Ausgabe von Raumbelegungsplänen für alle Hörsäle, Seminarräume und PC-Pools in denen Lehrveranstaltungen stattfinden. Die Auswahl erfolgt in der Dozenten- und Raumauskunft im Bereich Raumauskunft (Abbildung 19). Die Auflistung der darstellbaren Räume ist nach Gebäuden und Etagen geordnet. Der vom Benutzer ausgewählte Raum wird dann im Browserfenster angezeigt (Abbildung 21). Die Anzeige der Raumbelegungspläne erfolgt nach dem Schema der zuvor beschriebenen Ausgaben. Zusätzlich zur Anzeige der Veranstaltungen wird in der Überschrift der Parameter ausgegeben und im Anschluss an den Raumbelegungsplan weitere Hinweise und der Navigationsblock angezeigt.



Fachhochschule Stuttgart - Hochschule der Medien  
Fachbereich Information und Kommunikation

**Raumbelegung**  
für Raum No. 011 (Standort Wolframstr.)

Tag / Zeit	Veranstaltung [Sigel]	Lehrkraft
mo 8-10	Bibliothekarische Dienstleistungen: <b>Bestandsorganisation und -vermittlung</b> [42] Gruppe: BMD 2	Maier
mo 10-12	Bibliothekarische Dienstleistungen: <b>Bestandsorganisation und -vermittlung</b> [42] Gruppe: BMD 1	Maier
mo 12-14	Informationsressourcen 1: <b>Seminar</b> [5] Gruppe: BMD 2abc	Hoffmann
mo 14-16	7.3 Jugendbibliothek und -infothek: <b>Jugendfilm/Jugendbilder in Funk und Fernsehen</b> [16]	Heidtmann
mo 16-18	7.2 Musikbibliotheken: <b>Klassik cross over Jazz - Die Mischung der Stile</b> [36]	Krueger

Abbildung 21: Raumauskunft: Ausgabe



## 8 Zusammenfassung und Ausblick

Das hier vorgestellte Auskunftssystem wurde auf den sehr speziellen Einsatzbereich des Fachbereichs IuK der Hochschule der Medien angepasst. Als Auskunftssystem für Lehrveranstaltungen stellt es dort insbesondere für die Studierenden eine beträchtliche Erleichterung bei der Zusammenstellung ihrer Stundenpläne dar. Durch die Erstellung einer XML-Repräsentanz des Stundenplans wurde eine Grundlage geschaffen, um aus einer Datei mehrere individuelle Ausgaben generieren zu können.

Über die Möglichkeit, neben Stundenplänen für Studenten, auch Veranstaltungspläne der Dozenten und Raumbelegungspläne erstellen zu können, sowie auf Grund der serverseitigen Verarbeitung der XML-Datei, wurde der Nutzerkreis und der Einsatzbereich gegenüber vergleichbaren Lösungen erhöht.

Für das Auskunftssystem bieten sich neben dem Einsatz als Informationssystem für Lehrveranstaltungen an einer Hochschule in leicht abgewandelter Form auch andere Einsatzbereiche an. Solche wären zum Beispiel:

- ein Veranstaltungs-Auskunftssystem in den verschiedensten Bereichen (z.B. Theater, kommunale Verwaltung);
- ein internes Terminauskunftssystem in Unternehmen.

Das dargestellte System bietet sich als solide Basis für eine Weiterentwicklung an. Mit Hilfe von XSL können auch Ausgaben für andere Medien erstellt werden, z.B. Vorlagen für die Erstellung des gedruckten Vorlesungsverzeichnisses oder Konvertierung der Ausgaben in PDF. In weiteren Entwicklungsstufen ist eine mögliche Erweiterung in Bezug auf die Einbeziehung der Einmalbelegungen oder das Eintragen von zusätzlichen Veranstaltungen vorstellbar.

## Anhang A: CD-ROM

Die Stundenplandatenbank, sowie die Dateien des in dieser Arbeit beschriebenen Systems sind auf der CD-ROM enthalten. Die folgende Tabelle gibt eine Übersicht über den Inhalt der CD-ROM.

Tabelle 4: Inhalt der beigefügten CD-ROM

Datei	Bemerkung
Datenbank\ <i>ns2001.mdb</i>	Stundenplandatenbank; enthält alle Abfragen und Makros.
public_html\ <i>bm.php</i>	Semesterauswahl für die Studiengänge BMB und BMD
public_html\ <i>bmmaster.php</i>	Semester- und Modulauswahl für den Studiengang Master BM
public_html\ <i>diplomalt.php</i>	Semester- und Modulauswahl für die Studiengänge IM, ÖB und WB
public_html\ <i>dozentauswahl.php</i>	Formular zur Dozentenauswahl
public_html\ <i>dozent.php</i>	Startseite der Dozenten- und Raumauskunft
public_html\ <i>dozent.xsl</i>	Stylesheet zur Erstellung der Veranstaltungspläne für Dozenten
public_html\ <i>fakultativ.xsl</i>	Stylesheet zur Erstellung der Übersicht fakultativer Lehrveranstaltungen
public_html\ <i>id.php</i>	Semester- und Modulauswahl für den Studiengang ID
public_html\ <i>index.php</i>	Startseite des Auskunftssystems
public_html\ <i>indiplom.php</i>	Semester- und Modulauswahl für den Studiengang IW
public_html\ <i>inmaster.php</i>	Semester- und Modulauswahl für den Studiengang Master IW
public_html\ <i>logo.php</i>	Schriftzug des Fachbereichs und Logo der HdM

Datei	Bemerkung
public_html\ <i>module.php</i>	Hinweiseite bei falscher Modulauswahl
public_html\ <i>plan.dtd</i>	DTD für <i>plan.xml</i>
public_html\ <i>plan.xml</i>	XML-Datei für die Stundenplanauskunft
public_html\ <i>plan.xsl</i>	Stylesheet zur Erstellung der Stundenpläne
public_html\ <i>raum.dtd</i>	DTD für <i>raum.xml</i>
public_html\ <i>raum.xml</i>	XML-Datei für die Dozenten- und Raumauskunft
public_html\ <i>raum.xsl</i>	Stylesheet zur Erstellung der Raumbelungspläne
public_html\ <i>raumauswahl.php</i>	Formular zur Raumauswahl
public_html\ <i>start.php</i>	Nutzersteuerung der Stundenplanauskunft
public_html\ <i>stud.php</i>	Startseite der Stundenplanauskunft
public_html\ <i>style.css</i>	Stylesheet zur Formatierung der HTML-Seiten
public_html 	Grafik zur Navigation
public_html 	Schriftzug des Fachbereichs
public_html 	Logo der HdM
public_html 	Grafik zur Navigation
public_html 	Grafik zur Navigation

# Literaturverzeichnis

## Printmedien

[Anderson et. al. 2000] Anderson, Richard et. al. (2000): *XML professionell*. 1. Auflage. Bonn: MITP-Verlag GmbH

[Bach 2000] Bach, Mike (2000): *XSL und XPath – verständlich und praxisnah. Transformation und Ausgabe von XML-Dokumenten mit XSL*. München u.a.: Addison-Wesley

[Born 2001] Born, Günter (2001): *Jetzt lerne ich XML*. München: Markt+Technik Verlag

[Fung 2001] Fung, Khun Yee (2001): *XSLT. Working with XML and HTML*. Erste Auflage. Boston u.a.: Addison-Wesley

[Harold/Mean 2001] Harold, Elliott Rusty / Mean, W. Scott (2001): *XML in a Nutshell. A Desktop Quick Reference*. Erste Auflage. Sebastopol u.a.: O'Reilly

[HBI 1997] *Studien- und Prüfungsordnung der Fachhochschule Stuttgart – Hochschule für Bibliotheks- und Informationswesen*. In: *Das Studium an der Fachhochschule Stuttgart – Hochschule für Bibliotheks- und Informationswesen*. Fachhochschule Stuttgart – Hochschule für Bibliotheks- und Informationswesen. Stuttgart, 1997. S. 2-24

[HdM 2001a] *Studien- und Prüfungsordnung der Fachhochschule Stuttgart – Hochschule der Medien für die Diplomstudiengänge und Bachelorstudiengänge des Fachbereichs Information und Kommunikation*. In: *Studienführer und Vorlesungsplan. Wintersemester 2001/2002*. Fachhochschule Stuttgart – Hochschule der Medien. Stuttgart, 2001. S. 162-198

[HdM 2001b] *Studien- und Prüfungsordnung der Fachhochschule Stuttgart – Hochschule der Medien für die Masterstudiengänge Bibliotheks- und Medienmanagement und Informationswirtschaft*. In: *Studienführer und Vorlesungsplan. Wintersemester 2001/2002*. Fachhochschule Stuttgart – Hochschule der Medien. Stuttgart, 2001. S. 199-213

[Hütter 2001] Hütter, Bernhard (2001): *Bachelor- und Diplomstudiengänge. Bibliotheks- und Medienmanagement. Wahlpflichtbereiche und Module*. (zusammengestellt von B. Hütter / Stand: 10.07.01).

Fachhochschule Stuttgart – Hochschule für Bibliotheks- und Informationswesen. (unveröffentlichtes, internes Dokument)

**[McLaughlin 2001]** McLaughlin, Brett (2001): *Java und XML*. Erste Auflage. Köln u.a.: O'Reilly

**[Mintert/Menge 2000]** Mintert, Stefan / Menge, Rainald (2000): *XML verpuppt. Aufbereitung mit Cocoon und Extensible Server Pages*. c't. Hannover, 2000. Nr. 10, S. 222-227

**[Stede/Koch 2000]** Stede, Manfred / Koch, Stephan (2000): *The Adaptive University Calendar*. In: Brusilovsky, Peter et. al. (Hrsg.): *Adaptive Hypermedia and Adaptive Web-Based Systems. International Conference, AH 2000, Trento, Italy, August 2000. Proceedings*. Berlin u.a.: Springer. S. 384-387

**[Ziegler 2001]** Ziegler, Cai (2001): *Weltendämmerung. XML und Datenbanken: Einblick in Taimino*. iX. Magazin für professionelle Informationstechnik. Hannover, 2001. Nr. 6, S. 56-61

## Onlinemedien

**[Bakken/Schmidt 2001]** Bakken, Stig Sæther / Schmidt, Egon (2001): *PHP Manual*.  
URL: <http://php3.de/manual/en/index.php> (Stand: 11. Oktober 2001)

**[Bray et. al. 2000]** Bray, Tim et. al. (Hrsg.) (2000): *Extensible Markup Language (XML) 1.0 (Second Edition)*. *W3C Recommendation 6 October 2000*.  
URL: <http://www.w3.org/TR/2000/REC-xml-20001006> (Stand: 11. Oktober 2001)

**[HdM 2001c]** Fachhochschule Stuttgart – Hochschule der Medien (2001): *Vorlesungsverzeichnis Wintersemester 2001/2002*.  
URL: <http://v.hdm-stuttgart.de/stundenplan.html> (Stand: 29. September 2001)

**[HdM 2001d]** Fachhochschule Stuttgart – Hochschule der Medien (2001): *Stundenplaninformationssystem der HdM Stuttgart Fachbereich 3*.  
URL: <http://v.hdm-stuttgart.de/~sailer/planinfo2001w.php> (Stand: 29. September 2001)

**[Le Hégaret/Wood 2001]** Le Hégaret, Philippe / Wood, Lauren (2001): *Document Object Model (DOM)*.  
URL : <http://www.w3.org/DOM/> (Stand : 24. Oktober 2001)

**[Stepken 1999]** Stepken, Guido (1999): *MySQL Datenbankhandbuch*

URL: <http://www2.rent-a-database.de/mysql/> (Stand: 1. Oktober 2001)

**[Universität Magdeburg 2001]** Otto-von-Guericke-Universität Magdeburg (2001): *VVZ online. Version 0.8.*

URL: <http://gandalf.cs.uni-magdeburg.de/vvz/vvz.xml> (Stand: 26. Oktober 2001)

## **Erklärung**

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Stuttgart, den 5. November 2001 .....